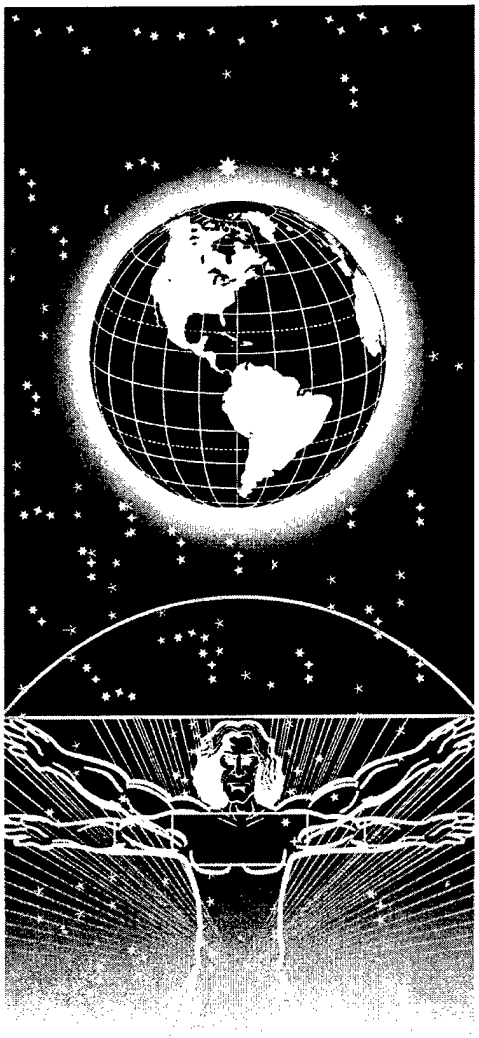
AFRL-HE-AZ-TR-1999-0249



UNITED STATES AIR FORCE RESEARCH LABORATORY

HEAD AND EYE MOVEMENTS IN FREE VISUAL SEARCH: EFFECTS OF RESTRICTED FIELD OF VIEW AND NIGHT VISION GOGGLE IMAGERY

George A. Geri
Paul A. Wetzel

Raytheon Training and Services Co.
6030 South Kent Street, Bldg 560
Mesa AZ 85212-6061

Elizabeth L. Martin

Air Force Research Laboratory
Warfighter Training Research Division
6030 South Kent Street, Bldg 558
Mesa AZ 85212-6061

December 1999

AIR FORCE MATERIEL COMMAND
AIR FORCE RESEARCH LABORATORY
HUMAN EFFECTIVENESS DIRECTORATE
WARFIGHTER TRAINING RESEARCH DIVISION
6030 South Kent Street, Building 558
Mesa AZ 85212-6061

NOTICES

Using Government drawings, specifications, or other data included in this document for any purpose other than Government-related procurement does not in any way obligate the US Government. The fact that the Government formulated or supplied the drawings, specifications, or other data, does not license the holder or any other person or corporation, or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

The Office of Public Affairs has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This report has been reviewed and is approved for publication.

ELIZABETH L. MARTIN
Project Scientist

DEE H. ANDREWS
Technical Director

JERRY L. STRAW, Colonel, USAF
Chief, Warfighter Training Research Division

Please do not request copies of this paper from the Air Force Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

Federal Government agencies and contractors registered with the Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
8725 John J. Kingman Road, Suite 0944
Ft. Belvoir, Virginia 22060-6218

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1999	Final - September 1998 to September 1999

4. TITLE AND SUBTITLE

Head and Eye Movements in Free Visual Search: Effects of Restricted Field of View and Night Vision Goggle Imagery

5. FUNDING NUMBERS

C - F41624-97-D-C-5000
PE - 62205F
PR - 2743
TA - B0
WU - 01

6. AUTHOR(S)

George A. Geri Paul A. Wetzel * Elizabeth L. Martin

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Raytheon Training and Services Co.
6030 South Kent Street, Bldg 560
Mesa AZ 85212-6061

*Air Force Research Laboratory, 6030 S. Kent Street, Bldg 558; Mesa AZ 85212-6061

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Training Research Division
6030 South Kent Street, Bldg 561
Mesa AZ 85212-6061

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFRL-HE-AZ-TR-1999-0249

11. SUPPLEMENTARY NOTES

Air Force Research Laboratory Contract Monitor: Dr Elizabeth L. Martin, AFRL/HEA (480) 988-6561 X-111, DSN 474-6111

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Combined head and eye movements were measured as observers searched for simple grating targets (2° or 6° dia.) superimposed on each of two levels of background detail, using each of three instantaneous fields of view (IFOV) (IFOV = 10°, 20°, or 40°). The relevant data were the pattern of head movements and the magnitude, direction, and fixation duration of gaze saccades. In addition, a separate set of observers performed the same search task using real night vision goggles (NVGs) (IFOV = 38°). It was found that increasing the IFOV resulted in a decrease in the magnitude of head movements and a concomitant increase in the magnitude of eye movements. This apparent tradeoff suggests that only about 10° of the visual periphery was effectively used in the visual search. Head scans and gaze magnitudes were independent of either the size of the test target or the level of background detail. Fixation duration was dependent on target size but only at the smallest IFOV. Saccadic duration was also plotted as a function of saccade magnitude (i.e., the main sequence), and we found that a power function with an exponent of about 0.35 gave the best fit to these data. Finally, the similarity of the data obtained in the IFOV = 40° condition and those obtained using real NVGs (IFOV = 38°) suggests that the conclusions reached here may be generalized to visual search performed with real NVGs under the stimulus conditions tested here.

14. SUBJECT TERMS

Eye movements; Field of view; FOV; Free visual search; Head and eye movements; Head movements; IFOV; Imagery; Instantaneous field of view; Night vision goggles; NVGs

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

PREFACE

The research described here was conducted by the Air Force Research Laboratory, Warfighter Training and Research Division (AFRL/HEA), under Work Unit 2743. The Work Unit Monitor was Dr. Elizabeth Martin. Support was provided by Raytheon Training and Services Co. under contract F41624-97-C-5000. The Laboratory Contract Monitor was Mr. Jay Carroll.

The authors thank Craig Vrana for providing the stimulus presentation software, and Lynnette Fischer for help in preparing the manuscript.

This page intentionally left blank

EFFECTS OF RESTRICTED FIELD OF VIEW AND NVG IMAGERY ON THE HEAD AND EYE MOVEMENTS USED IN FREE VISUAL SEARCH

1.0 Introduction

Under normal viewing conditions, visual search involves the use of both head and eye movements to direct the observer's gaze to objects of interest. Efficient visual search further requires that the relative contribution of the head and eye movements be adjusted in accordance with the viewing conditions, and with the spatial properties of the target and background. One major determinant of the head and eye movements used in visual search is the instantaneous field of view (IFOV) available to the observer. The IFOV is limited in many head-mounted displays, including night vision goggles (NVGs), and thus it might be expected that the characteristics of the head and eye movements used when performing visual search with these devices will be affected by this limitation (cf. Kotulak, 1992). The fact that observers depend on head movements to increase the field of view (i.e., by scanning the IFOV across the available visual field) has led to the suggestion that head scanning strategies be explicitly considered when head-mounted displays are used (Banks, Sternberg, Cohen, & DeBow 1971; Kotulak, 1992; Seagull & Gopher, 1997).

In the present study, we have measured both the head and eye movements of observers performing visual search using both simulated and real NVGs. The simulated NVG condition involved searching for a target presented on a background image using various IFOVs. In addition, the target size and level of background detail were varied to determine if these factors independently affected the characteristics of the head and eye movements used to perform the visual search. Among the head and eye movement characteristics measured were head-scan magnitude, period, and velocity, and gaze-saccade magnitude, direction, and fixation duration. These characteristics were also measured for observers searching for the same targets using real NVGs.

2.0 METHODS

2.1 Observers

A total of ten observers participated in the present study. Seven of the observers were tested in the instantaneous field of view condition. Two of those seven (GA and KL) and three additional observers (AA, BF, and CA) were tested using real NVGs. Observers AA, BF, and CA had training and operational experience using NVGs. All observers had uncorrected vision of 20/20 or better as determined by a Snellen chart.

2.2 Stimuli and Apparatus

The two background images that were searched are shown in Figure 1a. The 'desert' scene consisted of relatively low spatial detail, and thus was designated background-detail level one (BD1). The 'city' scene consisted of relatively high spatial detail and was designated BD2. The background images were standard photographs that were digitized and then displayed using only the green channel of a CRT projector (see below). The gray-scale histograms, as well as the brightness and contrast of the images were varied in an ad hoc manner to better approximate the visual characteristics of real NVGs, as judged by an experienced NVG user. The background images were derived from 1024 × 1024 × 8-bit images, which were displayed so as to subtend 68° horizontally and 52° vertically. The mean luminance of the background imagery was approximately 1.5 fL, and was equated at all points on the screen by appropriately scaling the luminance of the original digitized images as a function of the distance from their center.

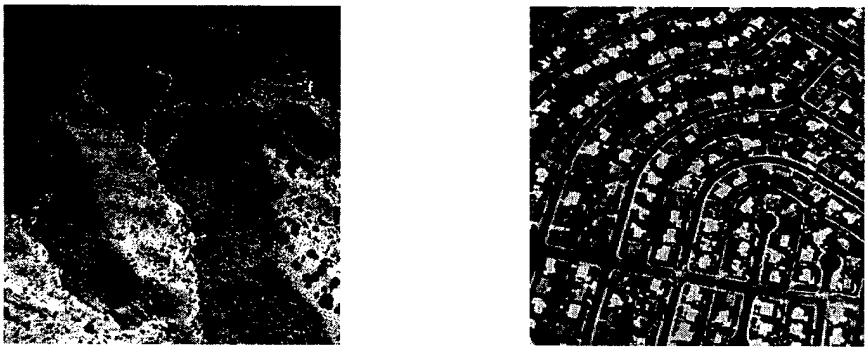
The observers' IFOV was determined by apertures placed approximately 1.5 inches in front of each eye and centered on each pupil. The aperture views were fused by the observers, giving the visual impression of a single aperture, of either 10°, 20°, or 40°, centered on the line of sight. The relative extent of the background images viewed through the three apertures is shown in Figure 1b.

The test targets for the search task were Gabor patches, which are sinusoidal gratings spatially localized using a gaussian envelope. The spatial frequency of the sinusoids was 3 cycles/target measured at the half height of the gaussian. The targets subtended either 2° or

6°, also measured at the gaussian half height. The full test stimuli extended to $\pm 2\sigma$ of their respective gaussian envelopes, and consisted of 32×32 and 96×96 pixels for the 2° and 6° test stimuli, respectively. Examples of the two test target sizes as they appeared at one position within the 'desert' background image are shown in Figure 1c. Test target contrast was varied by gradually changing the luminance (i.e., the grayscale value) of each background image pixel corresponding to the randomly chosen location of the test target. The luminance of a given pixel was varied between its original value, determined by its location in the background image, and its final value determined by its location in the target Gabor patch. The luminance of each pixel was changed so that the target reached its maximal contrast over a 15 s interval. In addition, the time between the start of a search trial and the beginning of the target onset interval varied randomly between 0 and 5 s.

All stimuli were displayed on a rear-projection screen using a Barco Model 801 CRT projector, and were viewed at a distance of 1.5 m. The background imagery was displayed and the test stimuli were controlled by a SGI Indigo Elan workstation. The source code used for test stimulus generation and presentation is given in Appendix A.

Head movements were measured in two-dimensions (pitch and roll) using a Polhemus 3Space headtracking system. Eye movements were measured using an El-Mar Vision 2000 eye tracking system. Head- and eye-movements were sampled at 120 Hz and all data collection was controlled by a PC. Special-purpose software was used for both real-time display of eye and gaze (i.e., head + eye) position, and for off-line data analysis (see below). ITT Night Vision Air Force F4949D night vision goggles with class B filters were used in the Real-NVG condition.

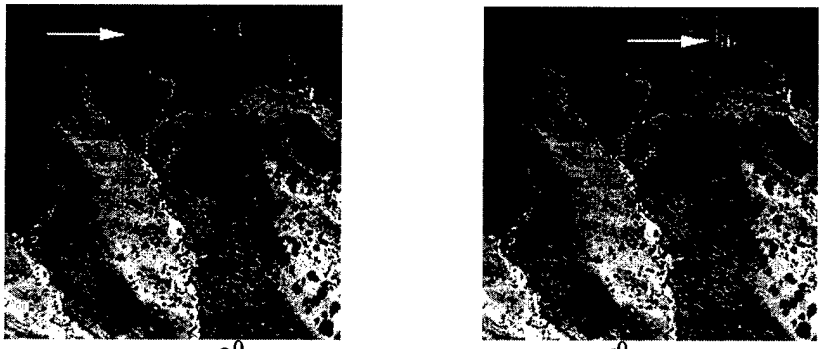


Desert (BD1) City (BD2)

a) Background Detail

10^0 20^0 40^0

b) Instantaneous Field-of-View (IFOV)



2^0 6^0

c) Target Size

Figure 1. Summary of the Variables Tested in the Present Study.

2.3 Procedure

The observers acclimated to the ambient light level of the experimental room by viewing one background image for 8-10 min while the head- and eye-movement systems were calibrated. The observers were then shown the test target for which they would be searching. They were informed that the target would initially be invisible and would then increase gradually in contrast until they indicated detection by pressing a mouse button. The observers were also informed that the target could appear anywhere within the background image. Each trial began when the observer was asked to begin searching for the test target.

Under the IFOV condition, all three IFOVs and both levels of background detail were tested in each experimental session. Two experimental sessions were run for each observer, one for each test stimulus size (either 2° or 6°). The order in which the conditions were tested was randomized for each observer. For each condition in a given session, the observers searched for the test stimulus in ten separate trials that lasted until either the test stimulus was detected or the allotted 35 s search time was exceeded. Each experimental session, including observer acclimation and equipment calibration, lasted about 45 min. All observers rested for 5-8 min after four conditions were run, and in addition they were allowed to rest as required between the other conditions.

Under the Real-NVG condition, four conditions (two test stimulus sizes and two levels of background detail) were run in a single experimental session that lasted about 30 min. All other experimental details were the same as under the IFOV condition.

2.4 Head and Eye Movement Recording

Horizontal and vertical head and eye movements were recorded throughout each 15-30 s experimental trial, at a sampling rate of 120 Hz. Data collection was automatically terminated after 35 s although observers were occasionally allowed to continue their search after data collection ended. Head movements were measured in two-dimensions (pitch and yaw) using a Polhemus 3Space Fast-Trak system. Changes in roll angle and three-dimensional head translation were found to be insignificant in the present experimental context. Eye movements were measured using an El-Mar Series 2000 eye tracking system.

Head- and eye-movements were sampled at 120 Hz and all data collection was controlled by a PC. Head data was acquired through the PC serial port while eye position data was acquired through two 12-bit A/D channels (Data Translation DT-2801A). All data acquisition was synchronized to the head-tracking system. Special-purpose software was used for both real-time display of eye and gaze (i.e., head + eye) position, and for off-line data analysis (see below).

2.5 Analysis of Head and Eye Movement Data

Horizontal and vertical gaze records were generated as a vector sum of the recorded head and eye movements. More than 95% of all data were collected as the observers scanned the visual display using relatively slow (< 35°/s) head movements. Under these conditions, compensatory eye movements were very accurate and we could detect no significant differences between the saccadic eye movements and their representation in the gaze record. We, therefore, restricted our analysis of saccades to the gaze record and have referred to those responses as 'gaze saccades'.

Gaze saccades in the individual horizontal and vertical records were identified and analyzed based on trend, velocity, and duration of movement. To qualify as the onset of a gaze saccade, the response for three successive sample points had to be in the same direction, and the change between each pair of successive points had to be at least 0.05°. The velocity at each sample point, n, in the full record, was found by measuring the average displacement of samples n-3 and n+3 and dividing that average by the time between those two samples. When the estimated velocity of a sequentially measured sample increased to 15 °/s, the previous sample was taken as the start of the saccade. Likewise, when the estimated velocity of a sample dropped below 15 °/s, it was taken as the end of the saccade. Any group of points that met both the displacement and velocity criteria was then identified as a component of a gaze saccade if its duration was between 17 and 108 ms, and if the fixation time associated with it was at least 50 ms.

Gaze saccade components identified in the horizontal and vertical records as described above were further analyzed to determine if they belonged to a pair and thus were part of a saccade that was not purely horizontal or vertical. The components were paired as

follows. The gaze saccade components in both the horizontal or vertical records were scanned in order to find the one that occurred first. The next component in the other record was then paired with it only if their onset times differed by less than 50 ms and their durations differed by less than 40 ms. If the components identified in the horizontal and vertical records had no correlate in the other record, they were taken to be purely horizontal or purely vertical gaze saccades.

3.0 RESULTS

3.1 Head, Eye, and Gaze Records

Shown in Figure 2 shows representative horizontal head and eye recordings, and the gaze record calculated from them, obtained from observer CS who was asked to search for the 2° target on the 'city' background image. The three graphs in the figure show records obtained for the (from top to bottom) 10°, 20°, and 40° IFOVs. The large-amplitude, smooth record in each graph corresponds to the observer's head movement, while the small amplitude records show the observer's eye movements. The large-amplitude, scalloped records are the observer's gaze which is the vector-sum of the corresponding head and eye records. Analogous data for the other observers tested in the IFOV condition are shown in Appendix B.

3.2 Head and Gaze Scanpaths

Scanpaths are the head (and or eye) movement patterns used by observers as they search a visual scene. The head (solid line) and gaze (dotted line) scanpaths obtained from the seven observers under one testing condition of the IFOV condition are shown in Figure 3. Analogous data for the five observers tested under the Real-NVG condition are shown in Figure 4. In both figures, the scanpaths correspond to one randomly selected trial of the ten trials run for each observer using the 2° target and the 'city' background (BD2) image. Although the scanpaths are noticeably different, most consisted of some combination of purely horizontal and vertical head movements. Further, both sets of data indicate that the

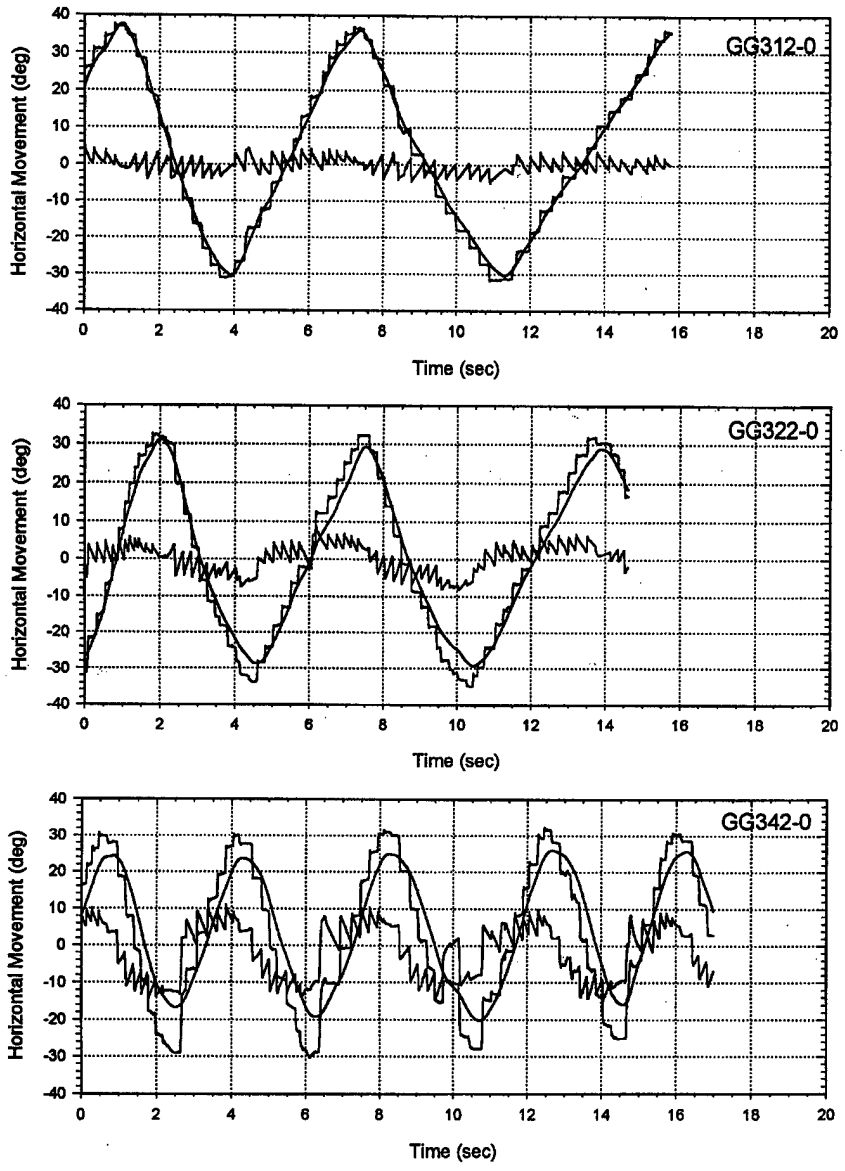


Figure 2. Examples of Horizontal Head, Eye, and Gaze Recordings

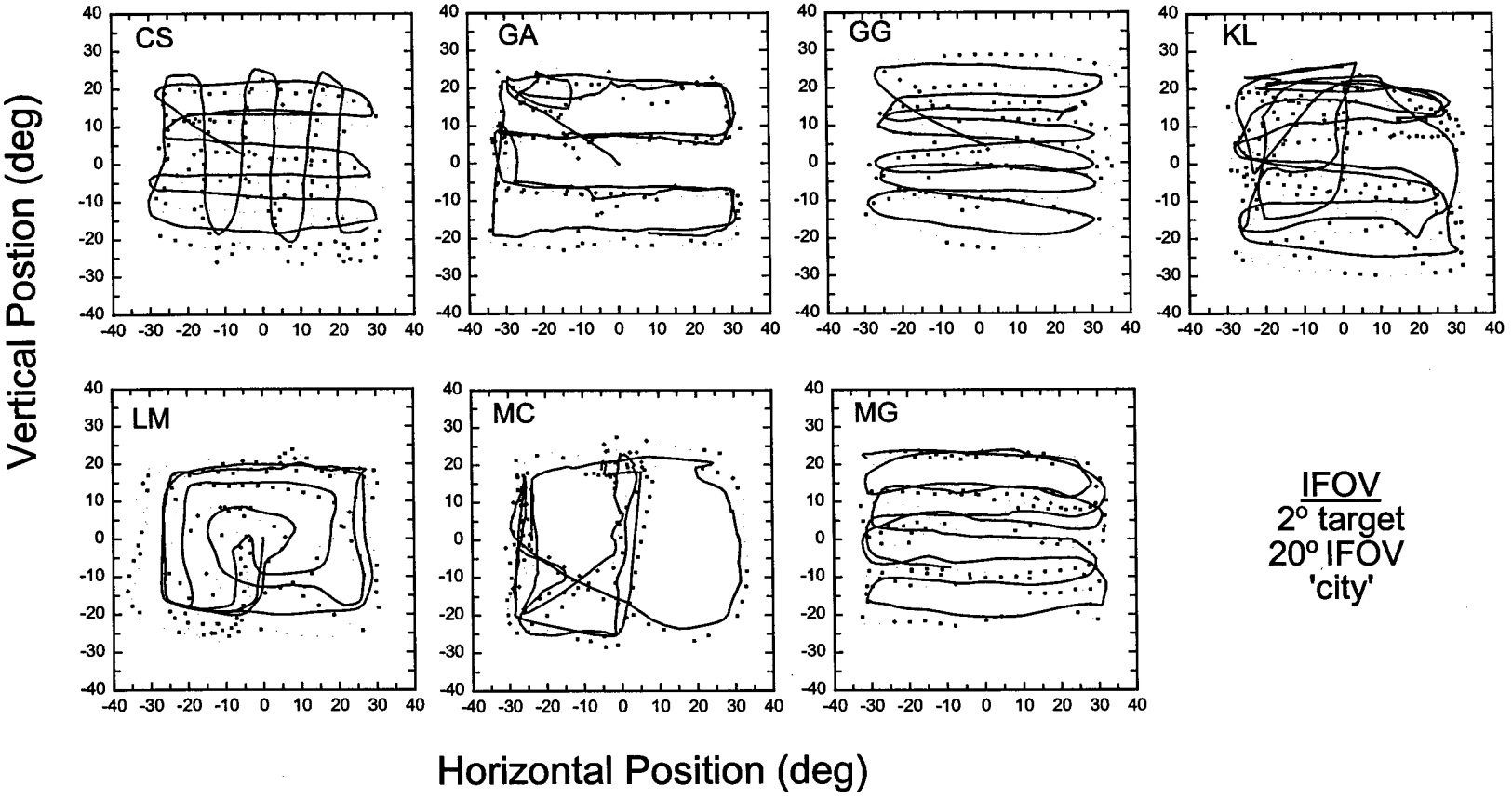


Figure 3. Head and Gaze Scanpaths for all Observers in the IFOV Condition.

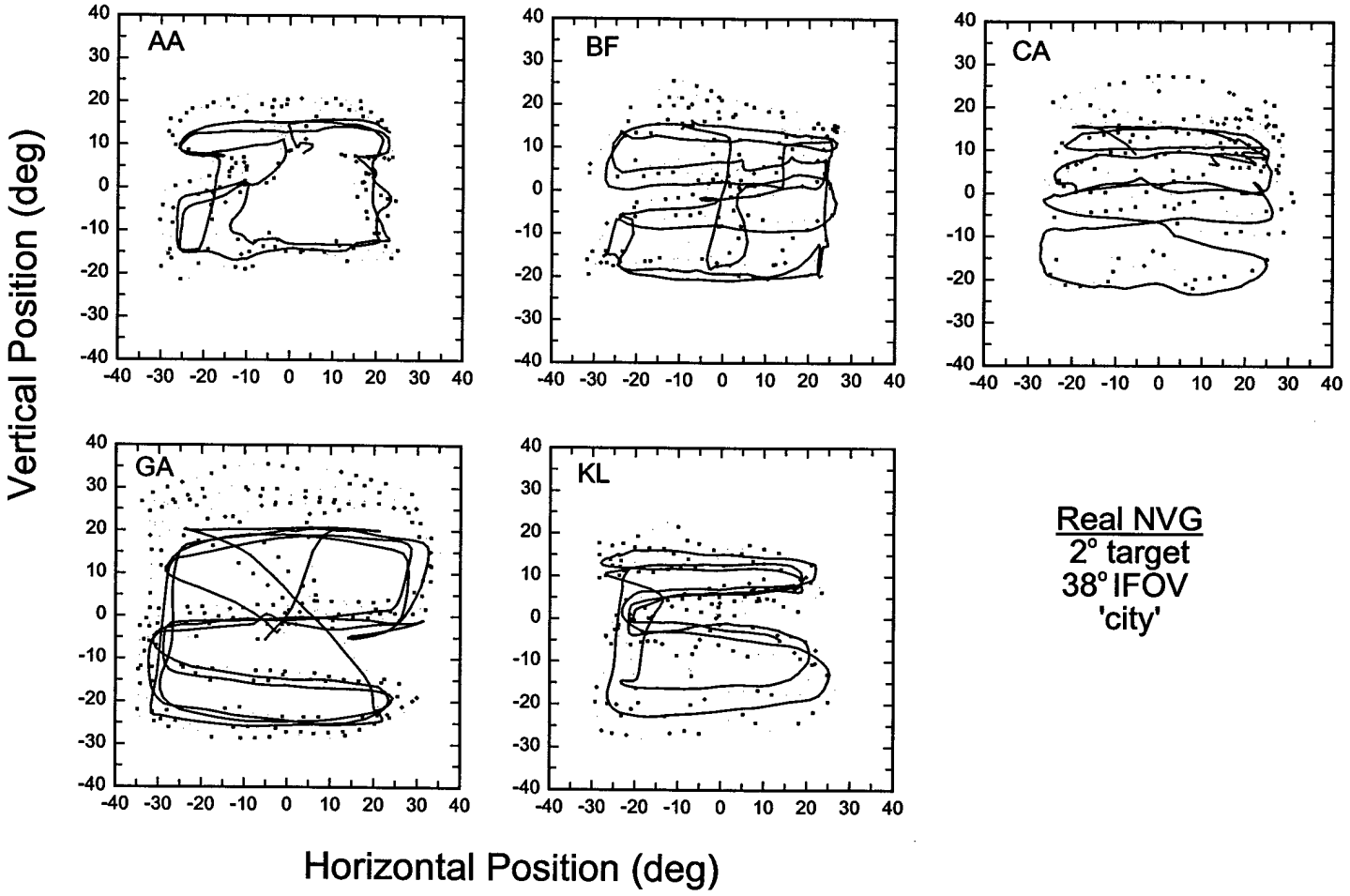


Figure 4. Head and Gaze Scanpaths for all Observers in the Real NVG Condition.

gaze scanpath, which includes eye position, deviated most from the head scanpath for the 40° IFOV. As can be seen in the data of Figure 5, obtained from observer GG, this was also true for the head and gaze scanpaths obtained from the other experimental conditions. Head and gaze scanpaths obtained from one randomly selected trial for all observers and all conditions tested in both the IFOV and Real-NVG studies are shown in Appendix C.

3.2.1 Head-Scan Magnitude

The data of Figs. 3-5, and of Appendix C, further indicate that the magnitude of the head scanpaths decreases in both the horizontal and vertical directions as the IFOV increases. This can be seen in Figure 6 where the magnitude of both the horizontal and vertical head scanpaths, averaged over all observers from which such data could be reliably obtained, is plotted as a function of the IFOV for each combination of target size and background image. The relative horizontal and vertical magnitudes are consistent with the relative size of the background image in the horizontal and vertical directions. For both the horizontal and vertical scanpaths, the decrease in scanpath magnitude with increasing IFOV is about the same under all conditions tested. Shown in Figure 7 is a comparison of the mean magnitudes of head scanpaths for the 40° IFOV condition and the Real-NVG condition. The data shown are for either the horizontal or vertical scanpaths obtained using either the 2° or 6° target presented on the less detailed ('desert') background image. The scanpaths for the data corresponding to 40° IFOV condition can also compared to the Real-NVG scanpaths (IFOV = 38°). When this is done for the four data points (collapsed over observers in each case) obtained from the two target size and two background detail levels, the head-scan magnitudes for the IFOV condition were found to be significantly larger than those obtained for the Real-NVG condition (horizontal: $t = 6.53$, $p < 10^{-4}$, $df = 6$; vertical: $t = 2.99$, $p = 0.024$, $df = 6$).

3.2.2 Head-Scan Period

Scan period is a measure of how many scans the observers used to cover the visual scene that they were asked to search. A larger period indicates that relatively more scans were used. The filled symbols plotted in the upper graph of Figure 8 are mean horizontal scan-period data for all observers and all stimulus combinations tested in the IFOV condition. These data indicate

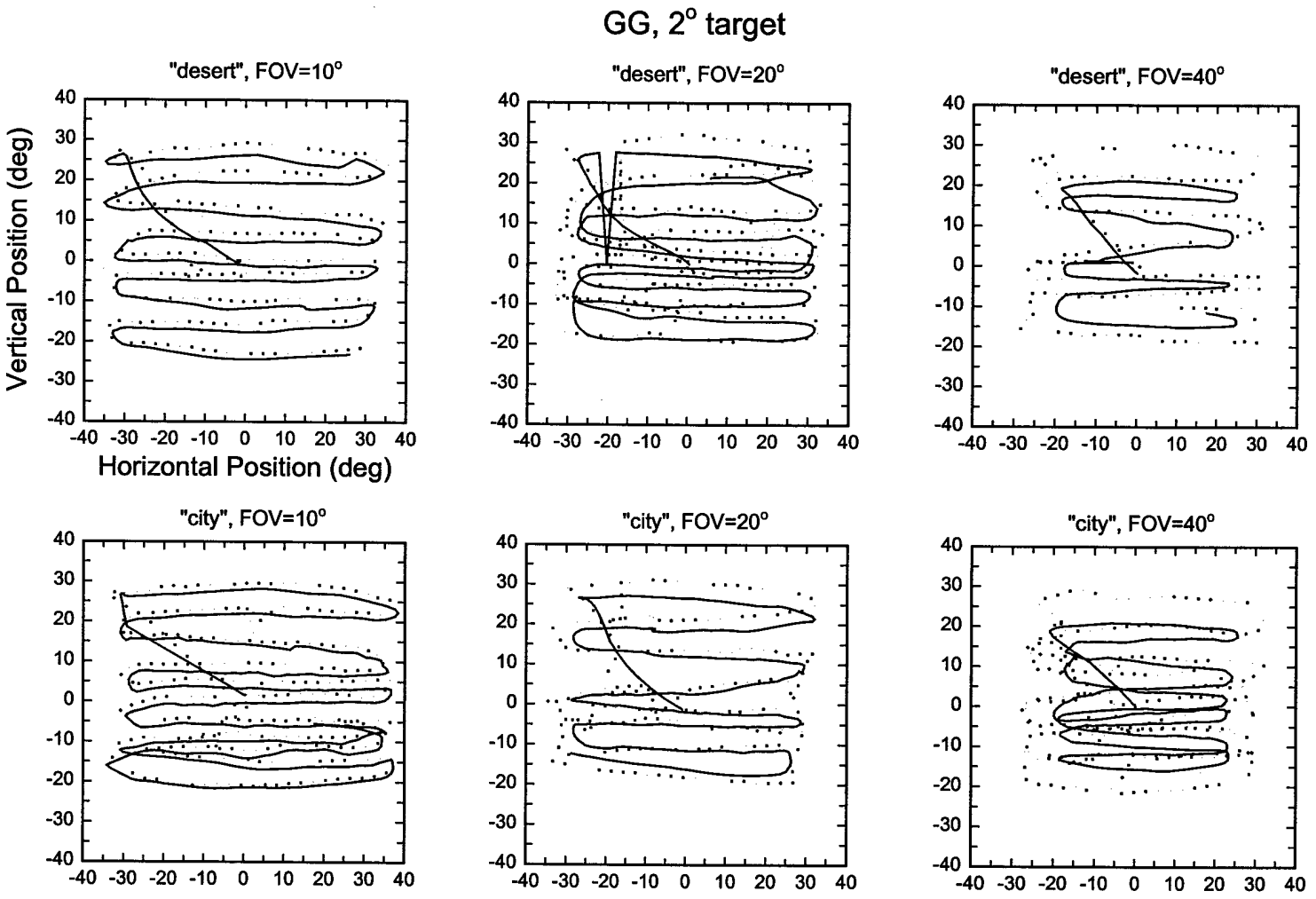


Figure 5. Head and Gaze Scanpaths for One Observer in the IFOV Condition.

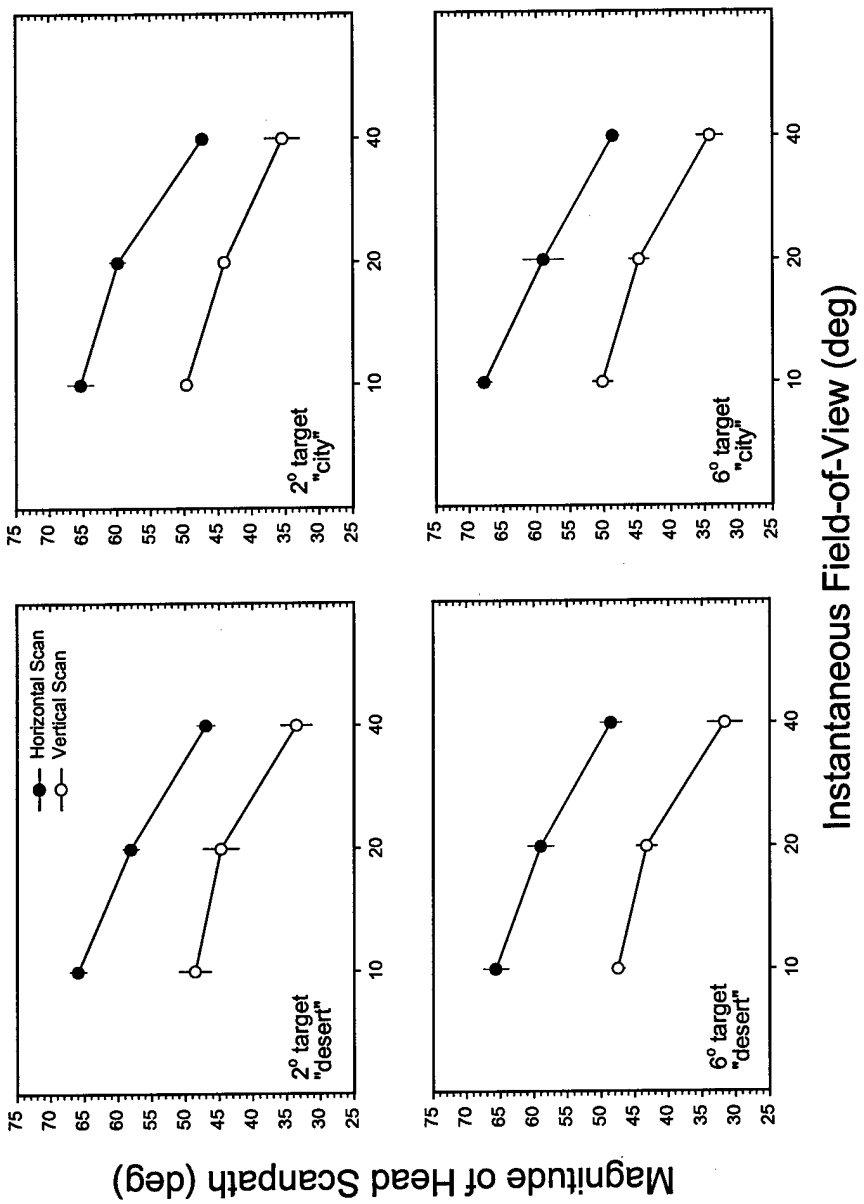


Figure 6. Magnitude of Head Scanpaths Averaged over All Observers in the IFOV Condition.

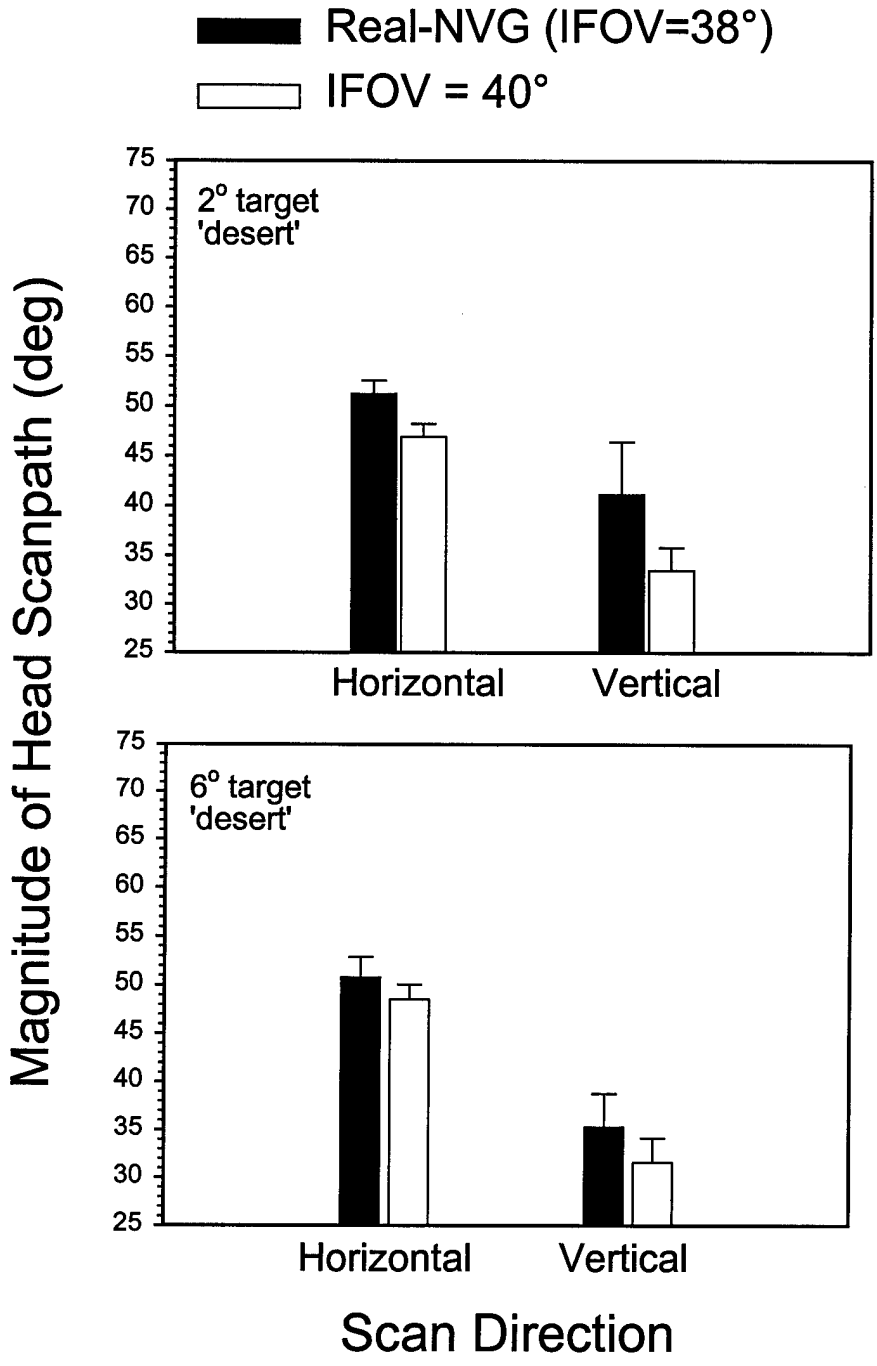


Figure 7. Head-Scan Magnitudes for the IFOV and Real-NVG Conditions

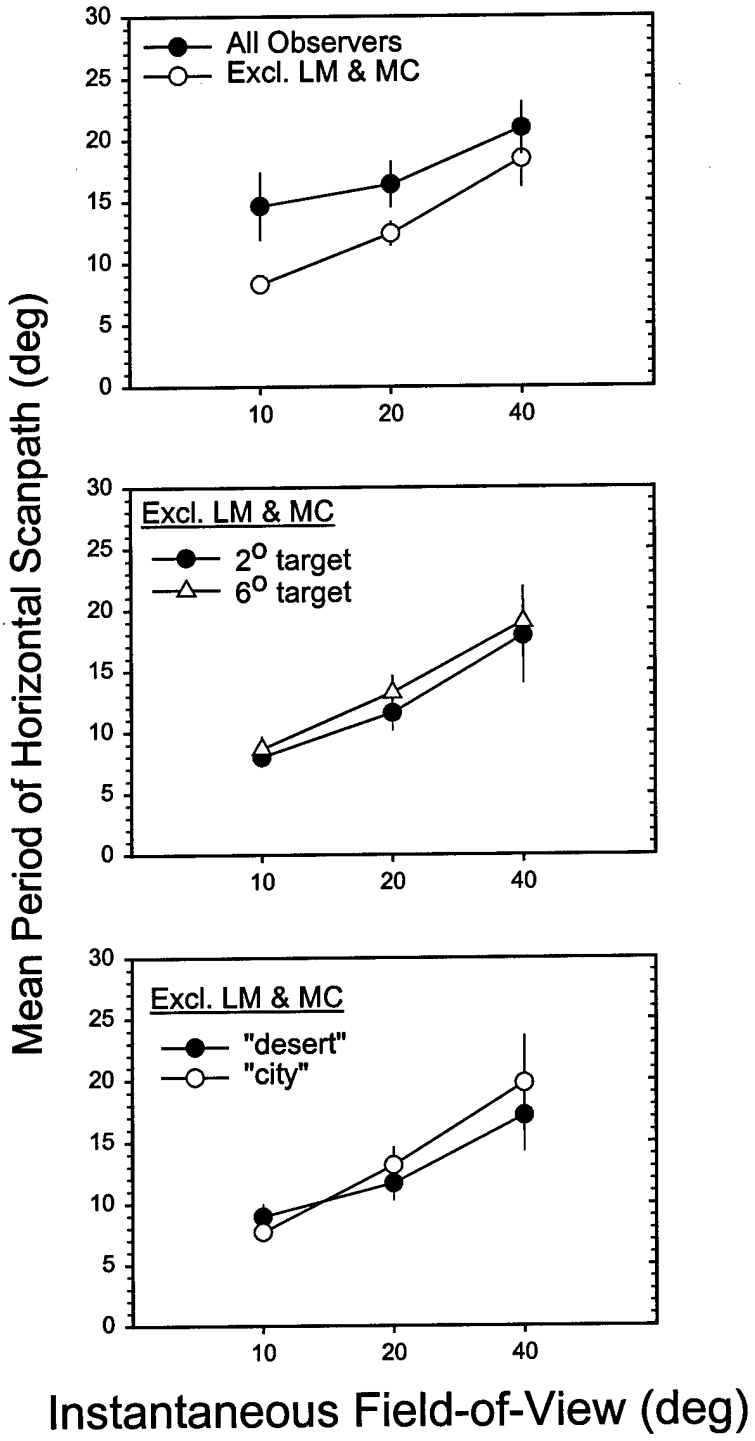


Figure 8. Head-Scan Period for the IFOV Condition.

that the scan period increased from about 14° to 20° as IFOV was increased. The individual horizontal scans of observers LM and MC either did not change much with IFOV or were somewhat difficult to identify. Therefore, the horizontal scan periods estimated after excluding the data of those two observers are shown as the open symbols in upper graph of Figure 8. After excluding observers LM and MC, scan period increased from about 8° to 18° as IFOV was increased. As may be seen from the center and lower graphs of Figure 8, the form of the function relating scan period to IFOV did not change significantly when the data were segregated on the basis of target size or level of background detail. Finally, the mean head-scan periods for the 40° IFOV and Real-NVG (IFOV=38°) conditions are compared in the three graphs of Figure 9. There was no significant difference in the data from the IFOV and Real-NVG conditions when the means (across observers) for the four target size and background combinations were compared ($t = 0.78$, $p = 0.47$, $df = 6$). There were not enough vertical scans to reliably estimate the vertical head-scan period.

3.2.3 Head-Scan Velocity

Head-scan velocity was measured throughout each of the single-trial scans from the IFOV condition, which are shown in Appendix C. These data are summarized in the three graphs of Figure 10. The upper graph compares the mean data from all observers with those obtained after excluding the data of observers MC and LM. The data of observer MC were excluded because her scanning strategy involved head movements that were more erratic, under certain testing conditions, than those of the other observers. The data of observer LM were excluded so that the velocity data could be compared to the scanpath period data (see Figure 8) from which her data were excluded for the reasons given earlier. The overall data shown in the upper graph of Figure 10 indicate that there was no significant difference between the data obtained from all observers and the data obtained after excluding observers LM and MC. Shown in the center graph of Figure 10 are the mean data (excluding MC and LM) separated by target size. Head-scan velocity was lowest for the 40° IFOV and highest for the 20° IFOV, although the differences among the three levels of IFOV were not statistically significant for either target size ($t < 1.48$, $p > 0.16$, $df = 18$). For each IFOV, the observers showed a higher head-scan velocity when searching for the

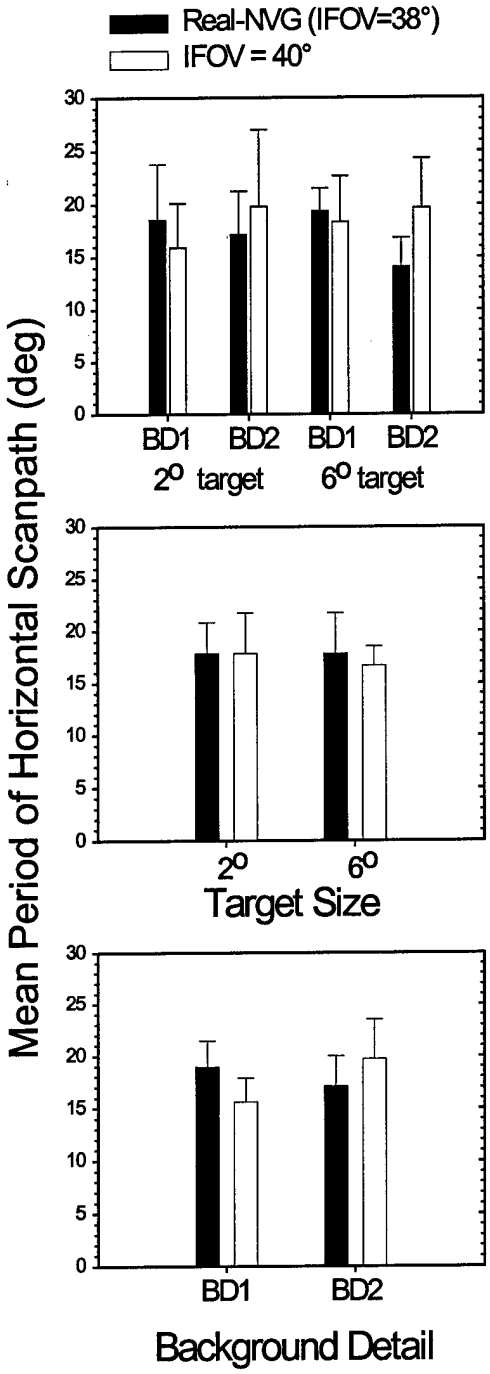


Figure 9. Head-Scan Period for the IFOV and Real-NVG Condition.

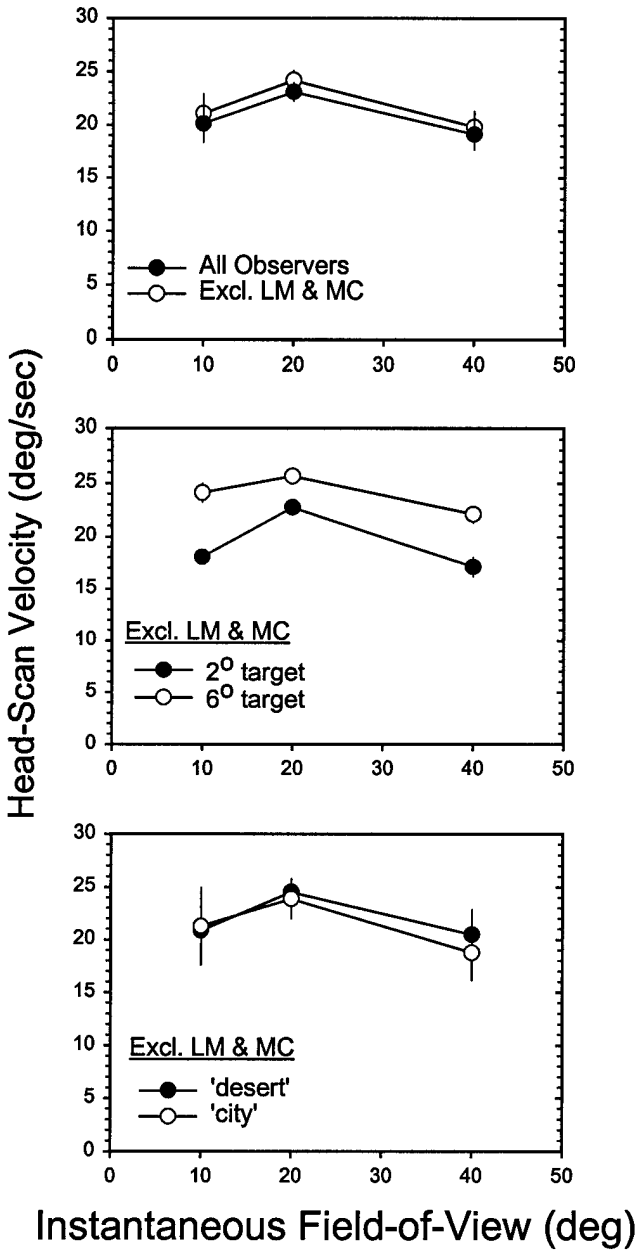


Figure 10. Mean Head-Scan Velocity for the IFOV Condition.

6° as compared to the 2° target ($t = 4.17$, $p = 10^{-4}$, $df = 58$). Finally, the mean head-scan velocity data separated by level of background detail (low: 'desert' vs. high: 'city') are shown in the lower graph of Figure 10. There is no significant difference in the head-scan velocities obtained for the two levels of background detail ($t = 0.43$, $p = 0.67$, $df = 58$).

3.3 Gaze-Saccade Magnitude

Gaze magnitude histograms for observer KL under all conditions tested in the IFOV condition, are shown in the four panels of Figure 11. Analogous data for all observers tested in the IFOV condition are shown in Appendix D, and all data are summarized in Table 1. The histograms include all saccades identified in all ten trials run under each experimental condition for each observer. Gaze magnitudes that exceeded the mean by more than $+3\sigma$ were excluded from the analysis. No responses smaller than the mean were excluded. The four panels of Figure 11 correspond to the four combinations of target size and level of background detail. Histograms for the 10°, 20°, and 40° IFOV levels are shown by the solid curve, the open circles, and the closed circles, respectively, in each panel. The gaze magnitude data are summarized in the graphs shown in Figure 12. In general, mean gaze magnitude is least for the 10° IFOV and increases approximately linearly with IFOV. There is no significant difference in the mean gaze magnitude for the two target sizes or two background-detail levels tested.

As can be seen in the histograms of Figure 11 and Appendix D, there was often a subpeak in the vicinity of 1°. Visual inspection of the eye movement records (see Figure 2) indicated that saccades less than or equal to about 1° generally occurred while head-scan velocity was changing, that is, at the beginning and end of the scans shown in the data of Figs. 3-5 and Appendix C.

Gaze magnitude histograms for the five observers tested using the 2° target and the 'desert' background image in the Real-NVG condition are shown in Figure 13. Under these testing conditions, the mean gaze magnitude varied from about 5.7° to 9.3°. The data for the three experienced NVG users are shown in the left column of Figure 13, and the data for the two inexperienced observers are shown in the right column of the figure. There was no significant difference in the mean gaze magnitudes measured for these two groups ($t = 1.55$, $p = 0.22$, $df =$

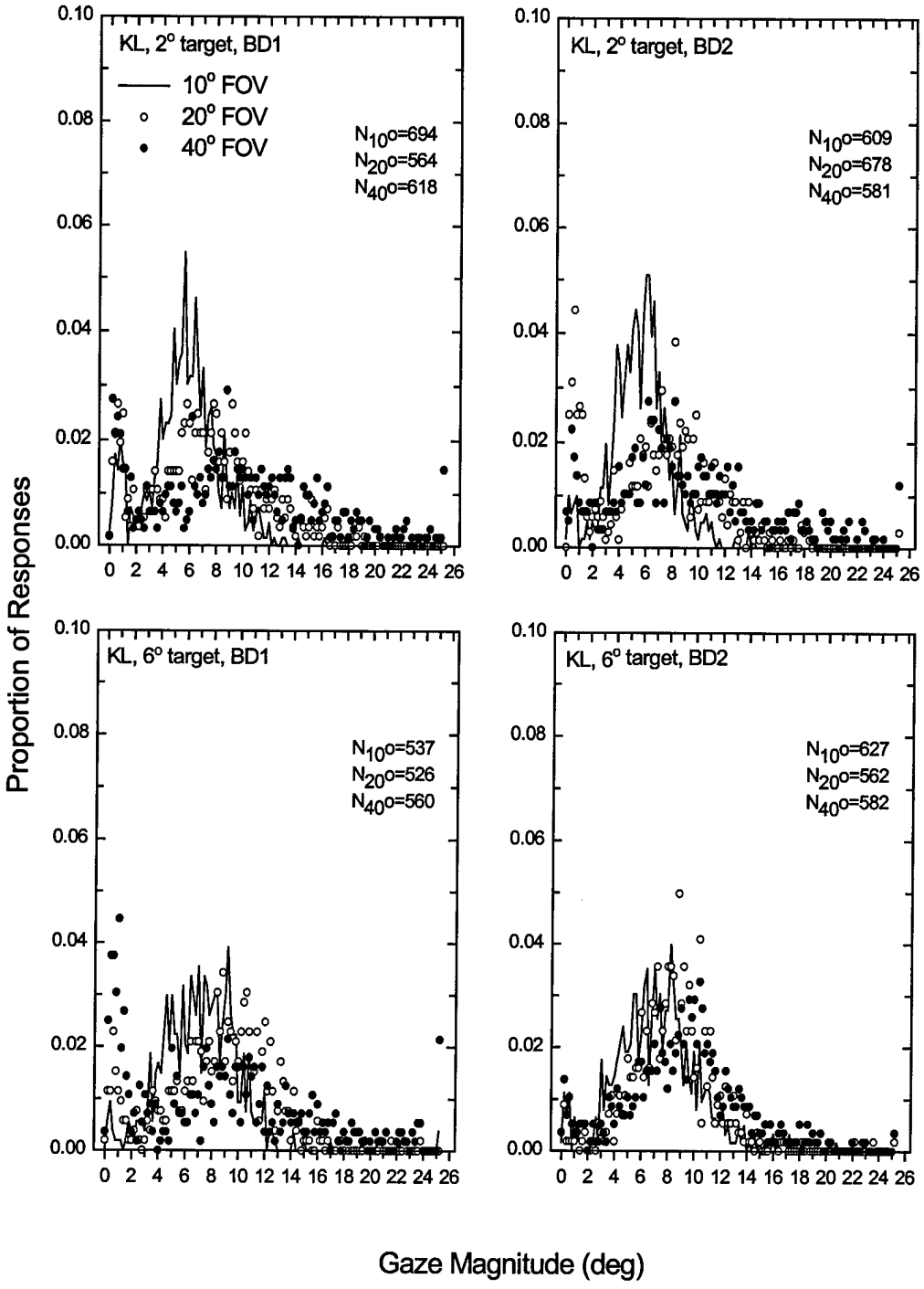


Figure 11. Gaze Magnitude Histograms for One Observer in the IFOV Condition.

Table 1.
Mean Gaze-Saccade Magnitudes (in deg) for all Observers in the IFOV Condition. Other entries are the standard error of the mean and the sample size.

FOV	BACK-GROUND	TARGE SIZE	Observers						
			CS	GA	GG	KL	LM	MC	MG
10^0	desert	2^0	4.87 2.46 928	6.41 3.64 709	5.78 2.02 1298	5.88 2.82 694	4.93 2.31 950	4.96 3.58 949	6.05 4.48 782
		6^0	5.72 3.55 836	5.97 3.05 776	5.74 2.33 1242	7.36 3.20 537	5.55 2.67 826	4.85 3.22 828	6.48 3.57 908
	city	2^0	3.86 2.18 911	5.22 3.78 750	5.75 2.50 1100	5.82 2.35 609	5.40 2.34 1017	4.16 3.58 907	5.75 4.09 873
		6^0	5.08 2.80 909	5.90 4.02 948	5.61 2.48 1206	7.26 3.04 627	5.12 2.39 922	4.69 3.60 783	7.56 3.84 918
20^0	desert	2^0	6.90 3.08 700	8.02 5.08 801	7.04 2.79 960	7.33 4.17 564	6.93 3.21 901	6.90 5.18 739	7.44 2.54 755
		6^0	7.04 4.02 713	7.90 5.51 586	7.06 3.27 916	8.54 4.03 526	5.80 2.63 870	6.76 4.25 727	8.03 4.68 813
	city	2^0	7.14 3.24 801	7.55 3.44 799	6.44 2.82 1203	6.98 3.78 678	6.60 3.19 877	6.27 4.33 658	7.36 2.85 721
		6^0	8.16 3.89 703	8.07 4.90 692	6.57 2.88 1077	8.51 3.23 562	5.78 2.86 818	7.54 4.85 870	7.61 4.59 875
40^0	desert	2^0	9.69 4.88 730	11.04 7.39 571	7.83 3.72 846	9.51 6.21 618	9.57 5.24 841	8.58 5.51 718	9.32 5.22 813
		6^0	9.04 3.96 547	8.79 7.15 583	7.94 3.50 802	8.72 7.33 560	9.57 5.87 812	8.52 5.91 688	9.97 5.21 706
	city	2^0	7.62 4.37 828	9.92 7.27 794	7.45 4.34 916	9.23 5.67 581	8.95 4.74 818	8.95 5.87 835	9.59 6.11 762
		6^0	9.46 3.91 562	8.89 6.94 718	8.55 4.04 936	9.31 4.28 582	9.12 5.06 843	8.36 5.76 721	10.01 5.25 597

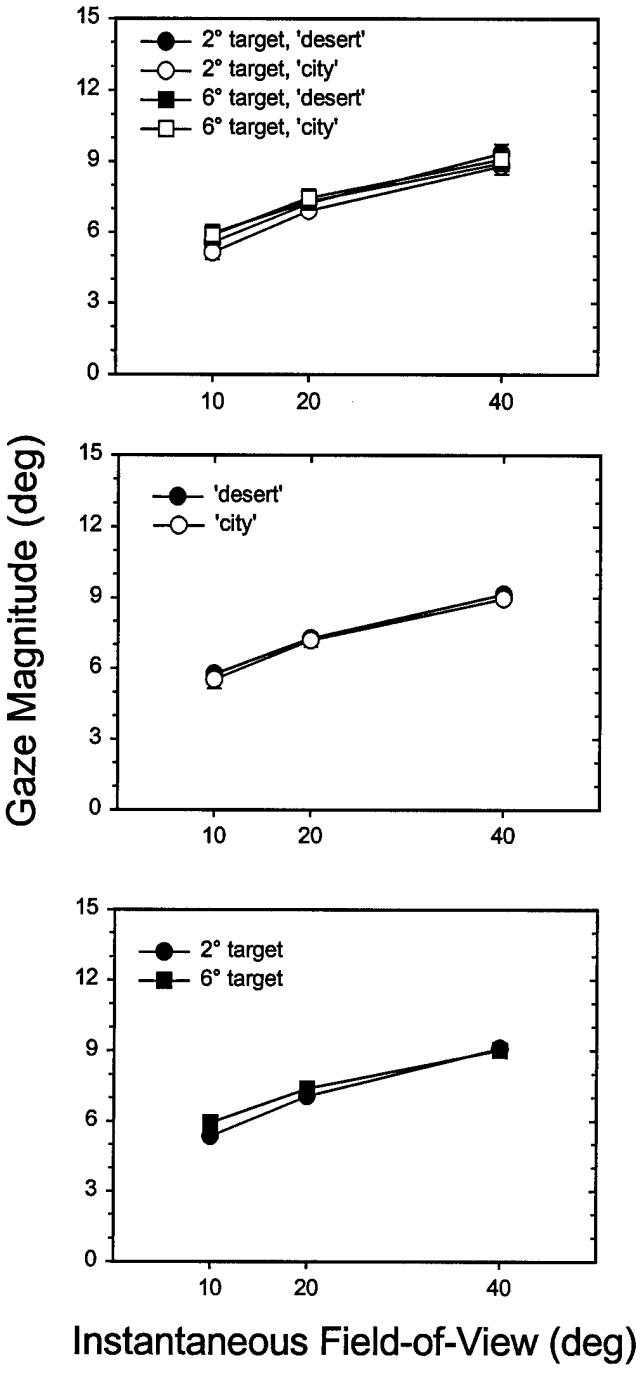


Figure 12. Mean Gaze Magnitude for the IFOV Condition.

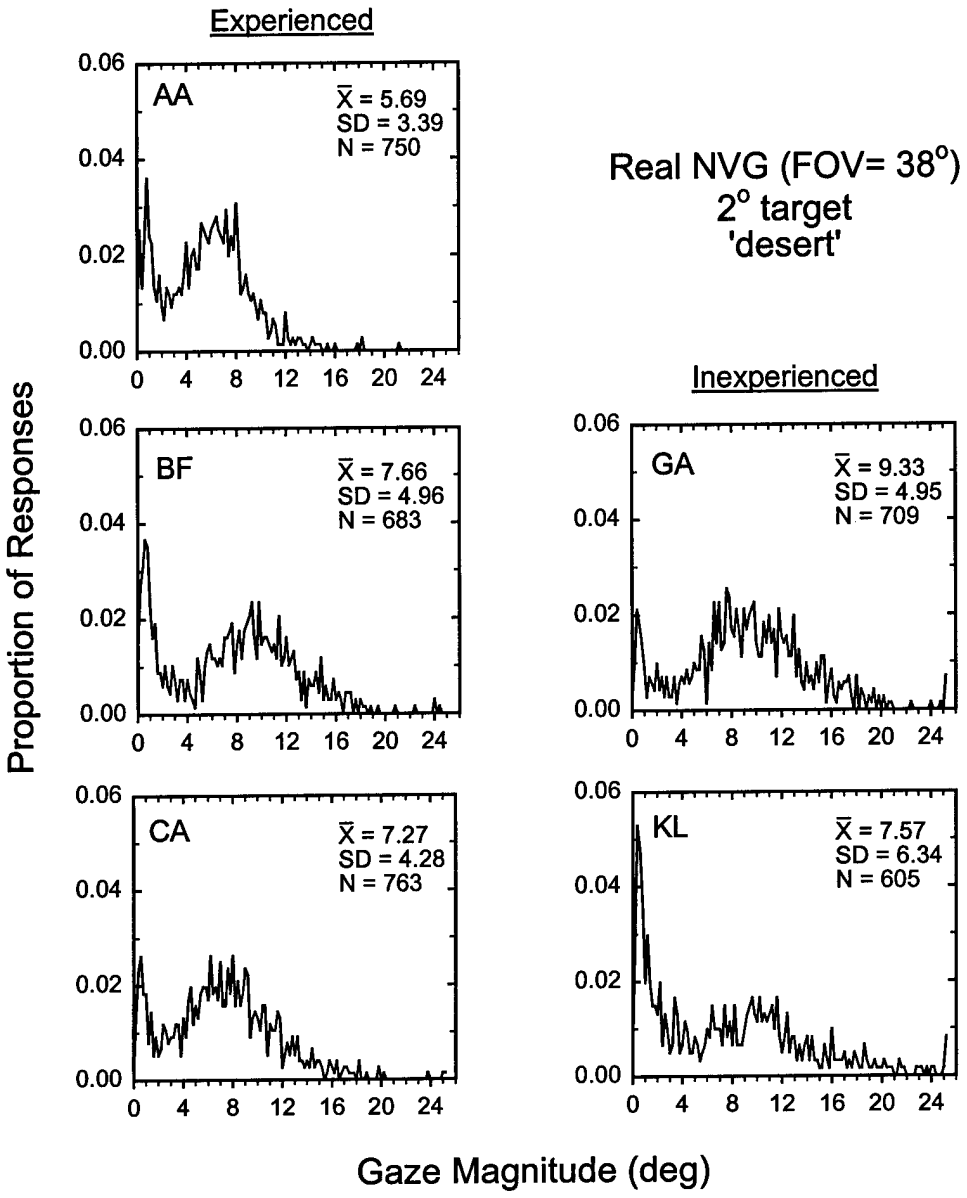


Figure 13. Gaze-Magnitude Histograms for the Real-NVG Condition.

3). The IFOV for the night-vision goggles used was about 38° and so roughly corresponded to the 40° level in the IFOV condition. Shown in Figure 14 are gaze magnitude histograms for the two observers (GA and KL) tested in both the IFOV/40° and Real-NVG conditions. For both observers, mean gaze magnitude was smaller under the Real-NVG condition (paired $t = 15.6$, $p = 0.04$, $df = 1$).

3.4 Gaze-Saccade Direction

Shown in Figure 15 are gaze-saccade direction histograms obtained from observer MG tested under the IFOV condition. The form of the histograms clearly shows the highly directional nature of the gaze saccades used by this observer while performing the present search task. The histograms all show major peaks near 0° and 180°, which correspond to leftward and rightward horizontal movements, respectively. The smaller peaks near 90° and 270° correspond to predominantly vertical saccades. The gaze direction histograms of all observers tested are shown in Appendix E. Five of the seven observers showed histograms similar to those shown in Figure 15, whereas two observers (LM and MC) showed responses that were much less directionally selective. The data for the observers tested under the Real-NVG condition are also shown in Appendix E, and are generally similar to those obtained under the IFOV condition.

3.5 Fixation Duration

Fixation duration histograms for the seven observers tested under the IFOV condition are shown for one stimulus set (2° target, 20° IFOV, and 'desert' background) in Figure 16. Analogous data for all stimulus conditions are shown for observer KL in Figure 17, and for all observers in Appendix F. The fixation-duration data from the IFOV condition are also summarized in Figure 18 and Table 2. The mean fixation durations for the seven observers vary from about 179–233 ms depending on the stimulus condition tested. As can be seen in the data of Figure 18, for the 2° test target, mean fixation duration for both levels of background detail is longer for the 10° IFOV than either the 20° or 40° IFOV ($t > 2.9$, $p < 0.03$, $df = 6$). There is, however, no significant change in fixation duration with IFOV for the 6° test target at either level of background detail ($t < 0.4$, $p > 0.7$, $df = 6$).

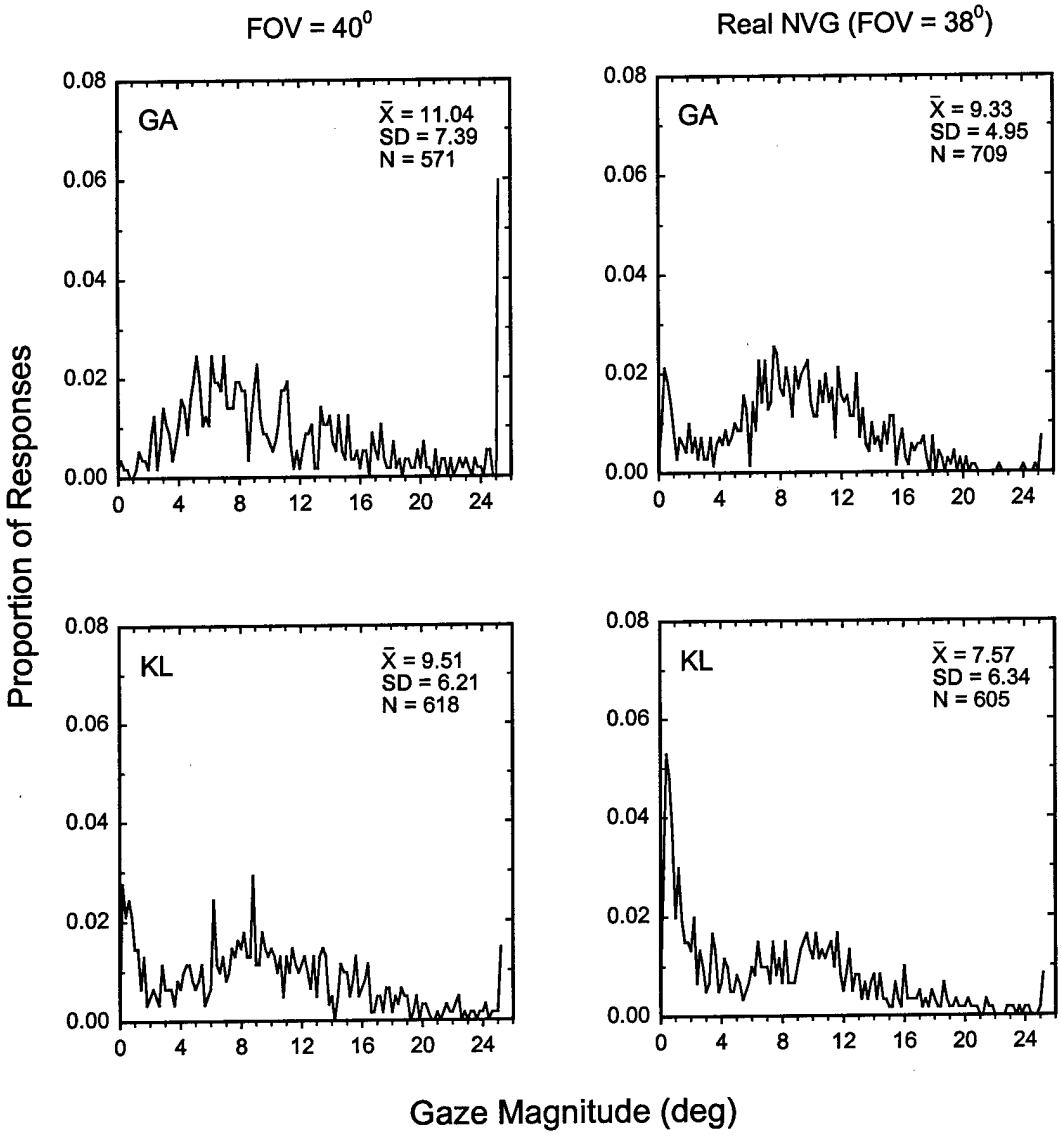


Figure 14. Gaze-Magnitude Histograms for the Two Observers Tested Under Boththe IFOV and Real-NVG Conditions.

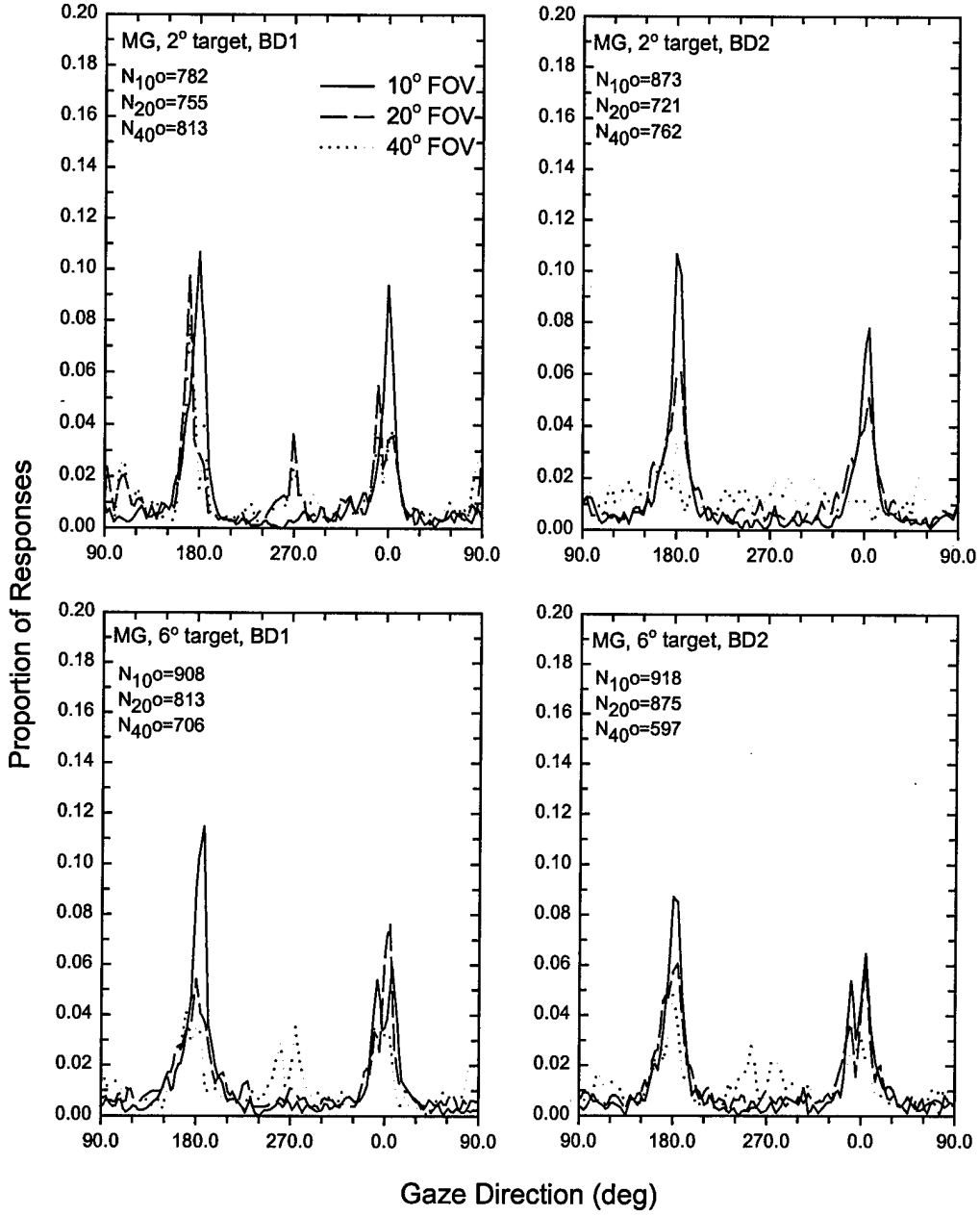


Figure 15. Gaze-Direction Histograms for One Observer in the IFOV Condition.

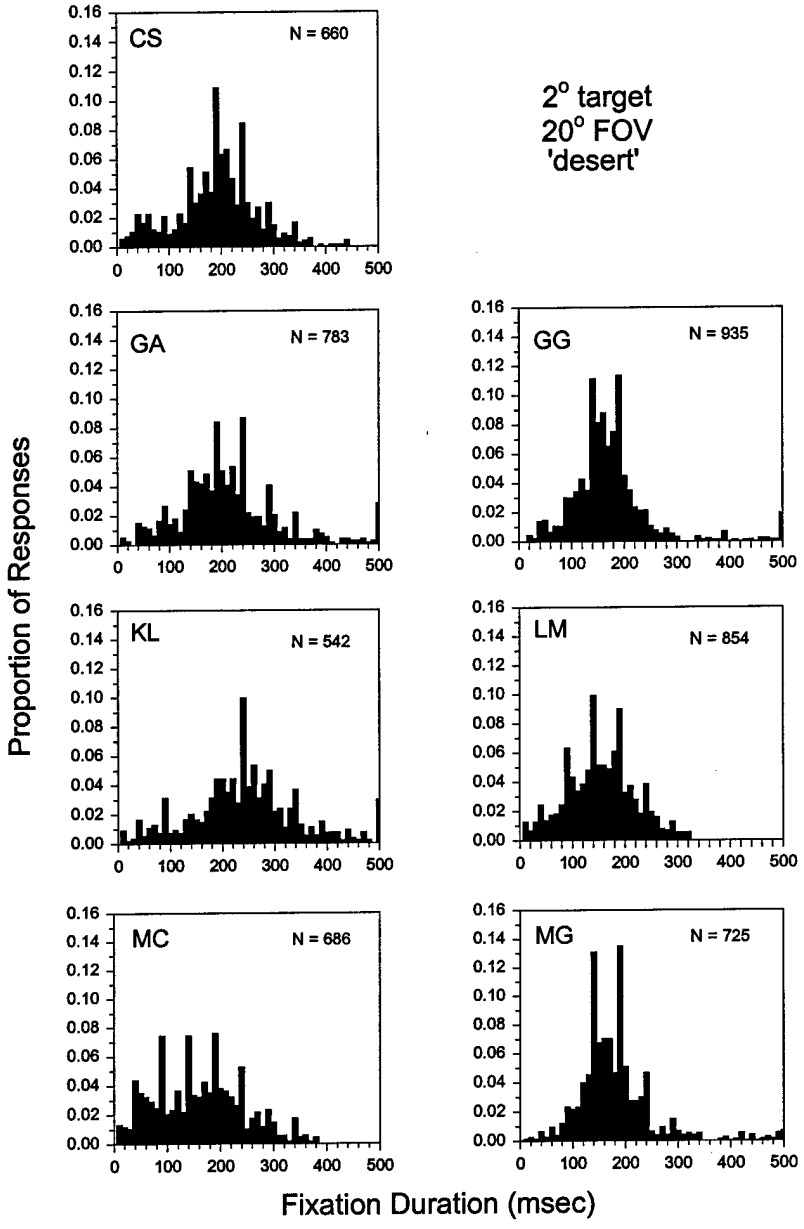


Figure 16. Fixation-Duration Histograms for the IFOV Condition,

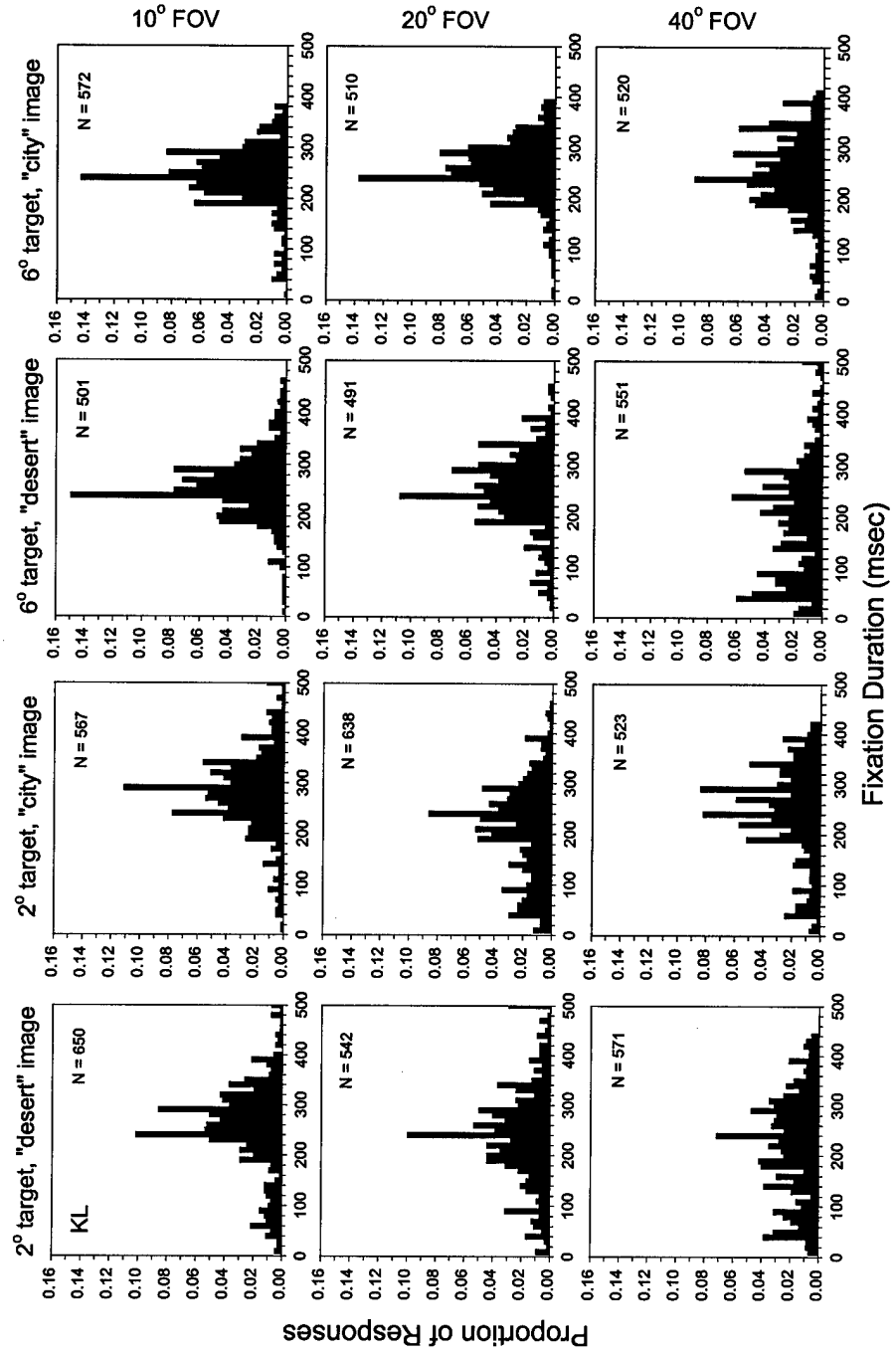


Figure 17. Fixation-Duration Histograms for One Observer in the IFOV Condition.

Table 2.
Mean Fixation Durations (in msec) for all Observers in the IFOV Condition. Other entries are the standard error of the mean and the sample size.

FOV	BACK-GROUND	TARGET SIZE	Observers						
			CS	GA	GG	KL	LM	MC	MG
10^0	desert	2^0	243.4 118.1 905	235.4 131.2 690	180.6 95.5 1263	259.0 92.0 650	243.4 118.1 905	229.1 108.1 900	210.1 94.1 751
		6^0	201.3 118.4 817	230.8 122.5 752	164.6 105.8 1204	263.2 64.7 501	159.7 74.7 783	182.1 84.9 779	157.4 61.4 846
	city	2^0	264.0 150.0 889	249.3 138.6 729	188.2 110.3 1069	285.5 83.5 567	184.3 68.7 955	258.7 129.6 852	201.4 85.4 830
		6^0	198.5 115.9 886	215.1 120.3 919	173.8 101.2 1178	246.7 60.6 572	169.1 78.4 893	215.8 96.7 719	132.7 56.2 867
20^0	desert	2^0	197.7 77.6 660	221.7 105.1 783	181.5 94.4 935	249.8 107.4 542	160.0 64.5 854	167.0 82.3 686	186.0 73.9 725
		6^0	167.0 83.7 680	206.5 111.5 570	167.3 68.0 892	255.6 76.9 491	185.6 72.7 816	169.3 74.0 672	154.7 66.3 763
	city	2^0	201.5 77.0 742	210.2 102.3 775	185.9 80.5 1161	212.0 95.5 638	169.0 67.0 828	199.2 75.4 592	197.4 62.0 651
		6^0	185.7 76.2 672	204.6 104.9 661	178.3 78.4 1052	261.6 57.0 510	185.8 68.5 759	169.1 67.3 795	164.1 69.0 822
40^0	desert	2^0	178.0 69.7 676	221.4 97.2 503	167.6 78.8 817	220.7 104.3 571	151.9 58.0 771	200.1 76.7 643	161.9 62.3 741
		6^0	206.7 79.4 524	212.0 112.8 540	163.0 74.6 773	198.4 117.4 551	133.4 63.3 775	194.2 92.8 620	142.1 60.9 666
	city	2^0	184.2 85.3 787	228.0 112.8 742	183.1 95.5 887	248.4 93.8 523	164.0 64.7 747	203.5 75.21 734	183.4 74.2 719
		6^0	208.2 75.2 529	203.0 114.1 668	148.4 67.2 908	253.6 80.9 520	154.7 64.0 790	203.3 86.1 647	160.9 64.6 542

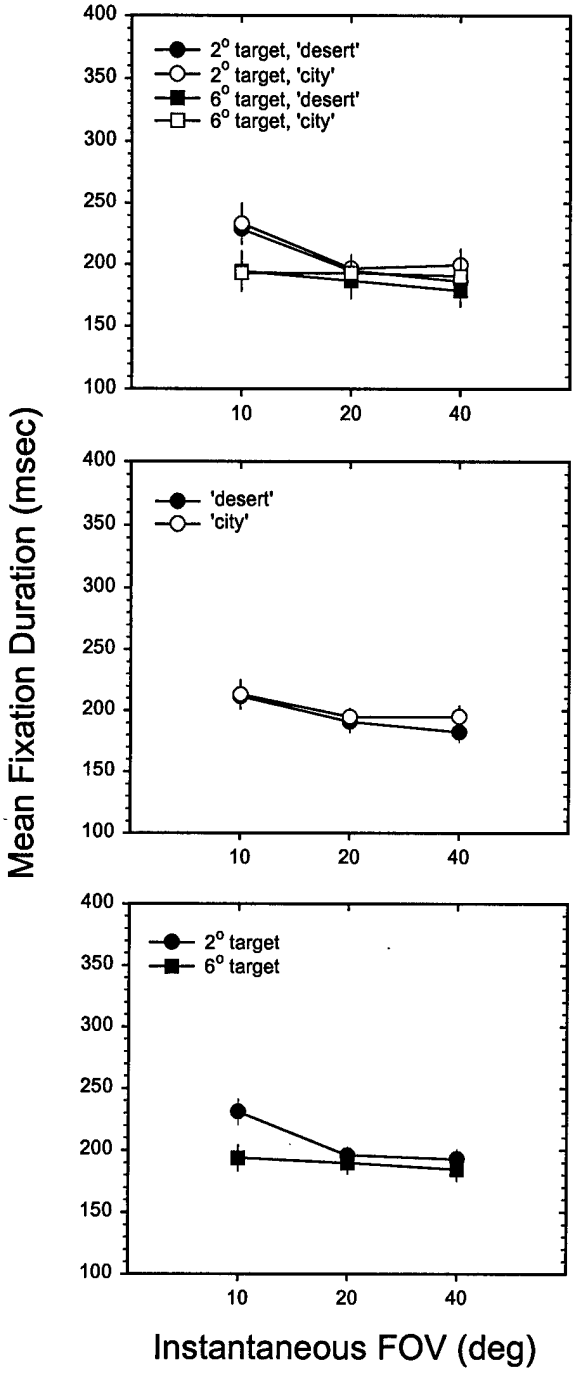


Figure 18. Mean Fixation Duration for the IFOV Condition.

Mean fixation duration for the five observers tested under the Real-NVG condition are shown for all stimulus conditions in Table 3. The means vary from about 183 to 264 ms depending on the condition tested, which is a slightly larger range than was found under the IFOV condition. The means for each observer collapsed across all testing conditions was compared with the means from the IFOV condition for the same four stimulus conditions and IFOV=40° (i.e., data rows 9-12 in Table 2). No significant difference was found in the mean fixation durations for the IFOV and Real-NVG conditions ($t = 1.58$, $p = 0.14$, $df = 10$).

3.6 Main Sequence Data

The relationship between gaze-saccade duration and magnitude is shown in Figure 19 for one stimulus combination (2° target, 20° IFOV, and 'desert' background) for all observers tested under the IFOV condition. Shown in Figure 20 are analogous data for all stimulus conditions for two representative observers (GG and MC). Finally, a comparison of the gaze saccade duration and magnitude data obtained from the IFOV and Real-NVG conditions are shown in Figure 21. These data were obtained by pooling all measurable responses from all seven observers in the IFOV condition and all five observers in the Real-NVG condition. The data of Figs. 19–21 were fitted with a power function of the form duration = $a \cdot (\text{magnitude})^b$. The best-fit parameters ranged from about 29.1–38.3 for parameter a, and from about 0.24–0.40 for parameter b. The associated R^2 values for the nonlinear regressions ranged from about 0.29–0.74 indicating the proportion of the variance in the data accounted for by the fitted function. In general, the differences in the parameter values for the various stimulus conditions within an observer are similar to the differences across observers. However, there is a significant difference, in both parameter values, for the data obtained under the IFOV and Real-NVG conditions ($t > 4.0$; $p < 0.007$; $df = 6$).

Table 3.
Mean Fixation Duration (in ms) for All Observers in the Real-NVG Condition. Other entries are the standard error of the mean and the sample size.

		Observers				
Background Detail	Target Size	AA	BF	CA	GA	KL
desert	2°	183.061 2.6783 750	227.287 4.078 683	237.848 3.882 763	190.759 3.750 709	227.602 4.740 605
	6°	188.913 3.655 619	205.962 4.655 501	211.756 4.198 500	195.006 4.611 528	257.513 6.045 458
city	2°	195.103 3.081 847	227.427 4.403 736	229.579 3.852 794	186.966 3.415 731	245.804 4.497 665
	6°	186.099 2.898 779	209.470 4.141 672	212.936 3.662 615	202.062 3.960 676	264.085 5.745 495

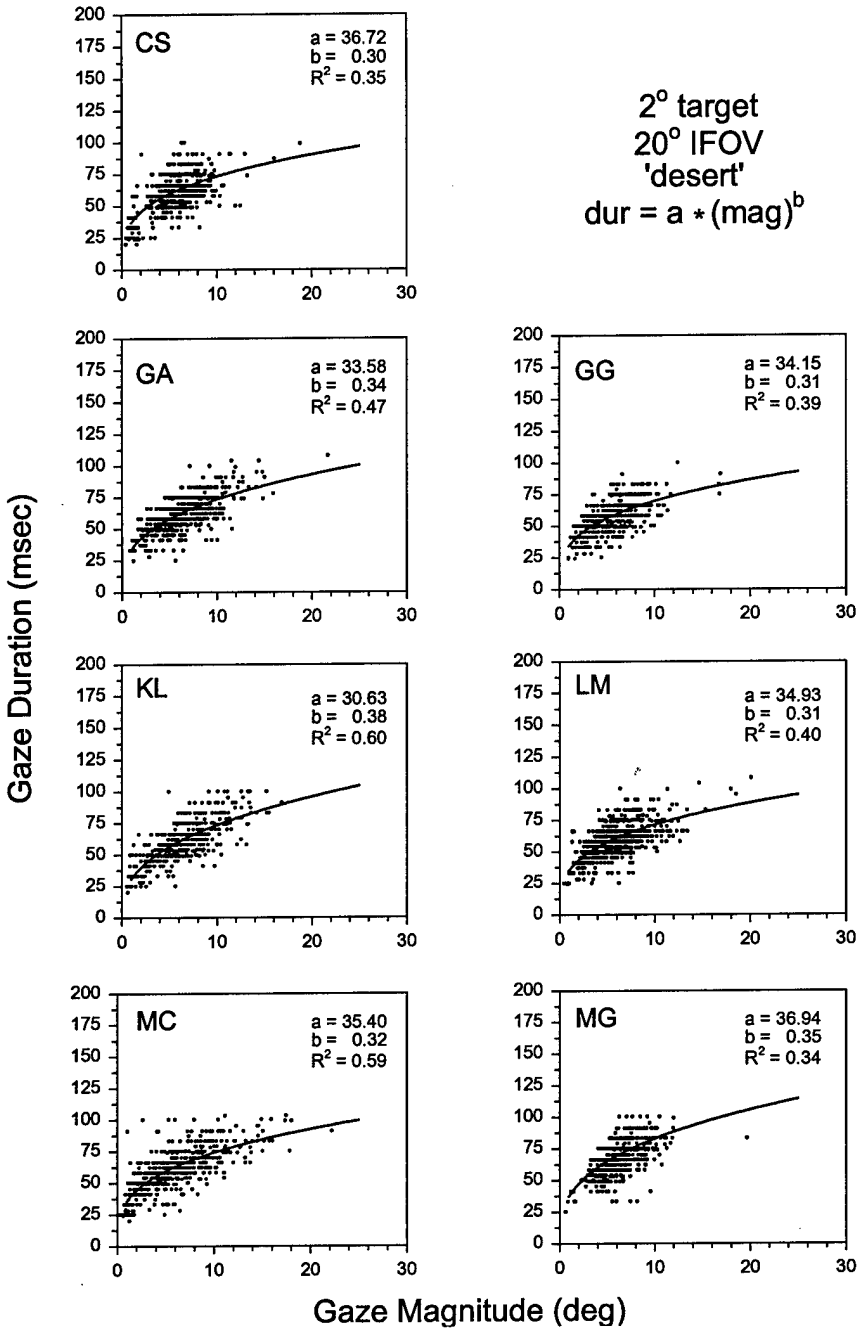


Figure 19. Main Sequence Data for All Observers in the IFOV Condition.

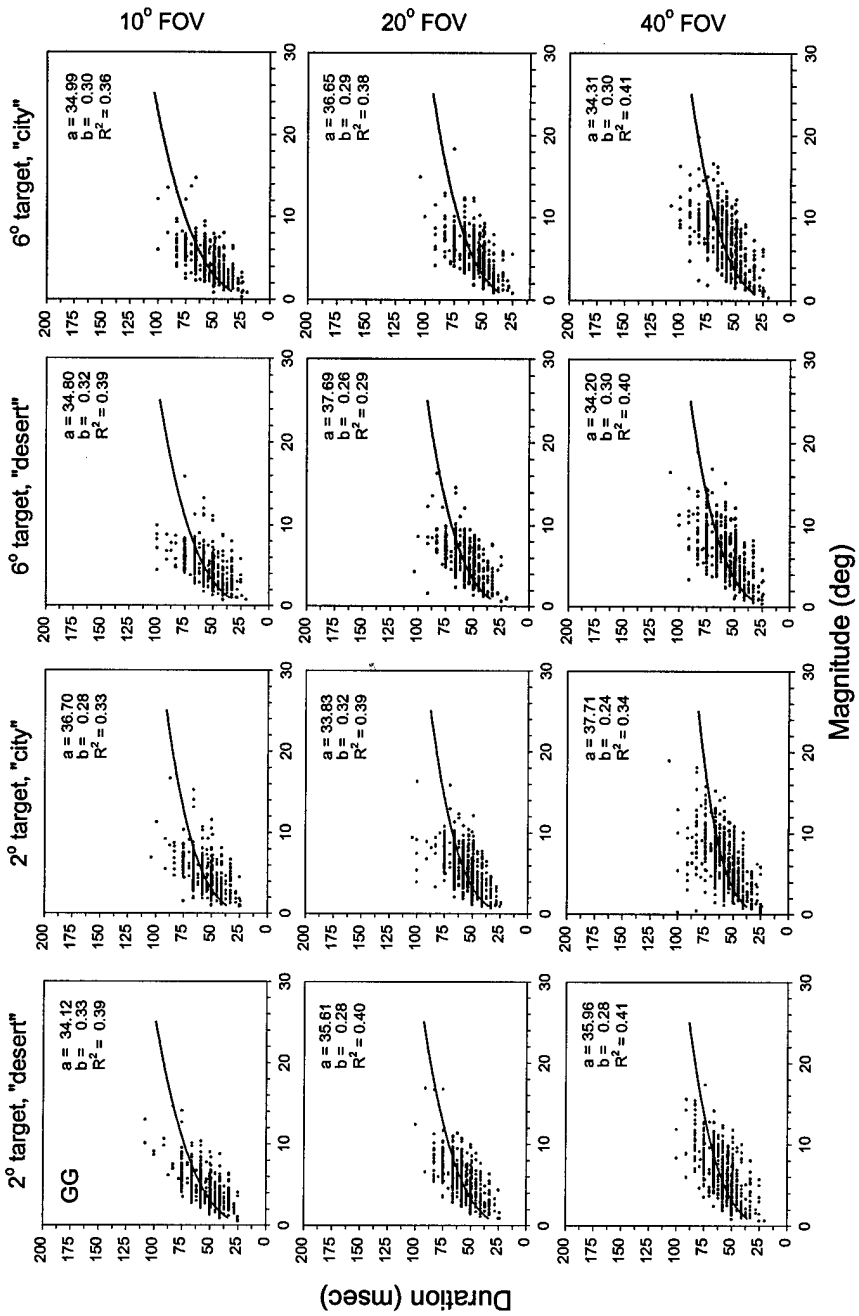


Figure 20. Main Sequence Data for One Observer in the IFOV Condition.

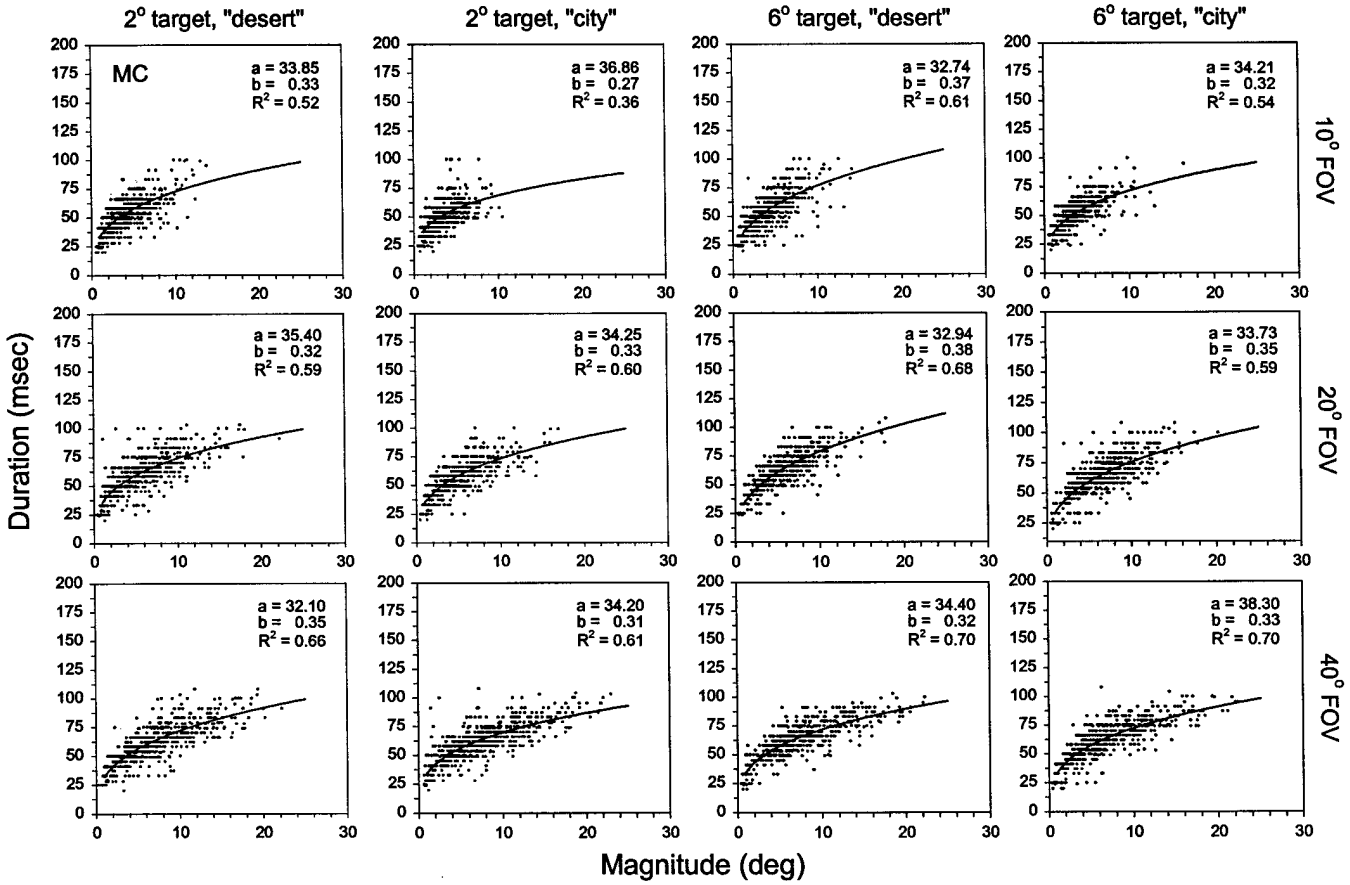


Figure 20.(con't.) Main Sequence Data for One Observer in the IFOV Condition.

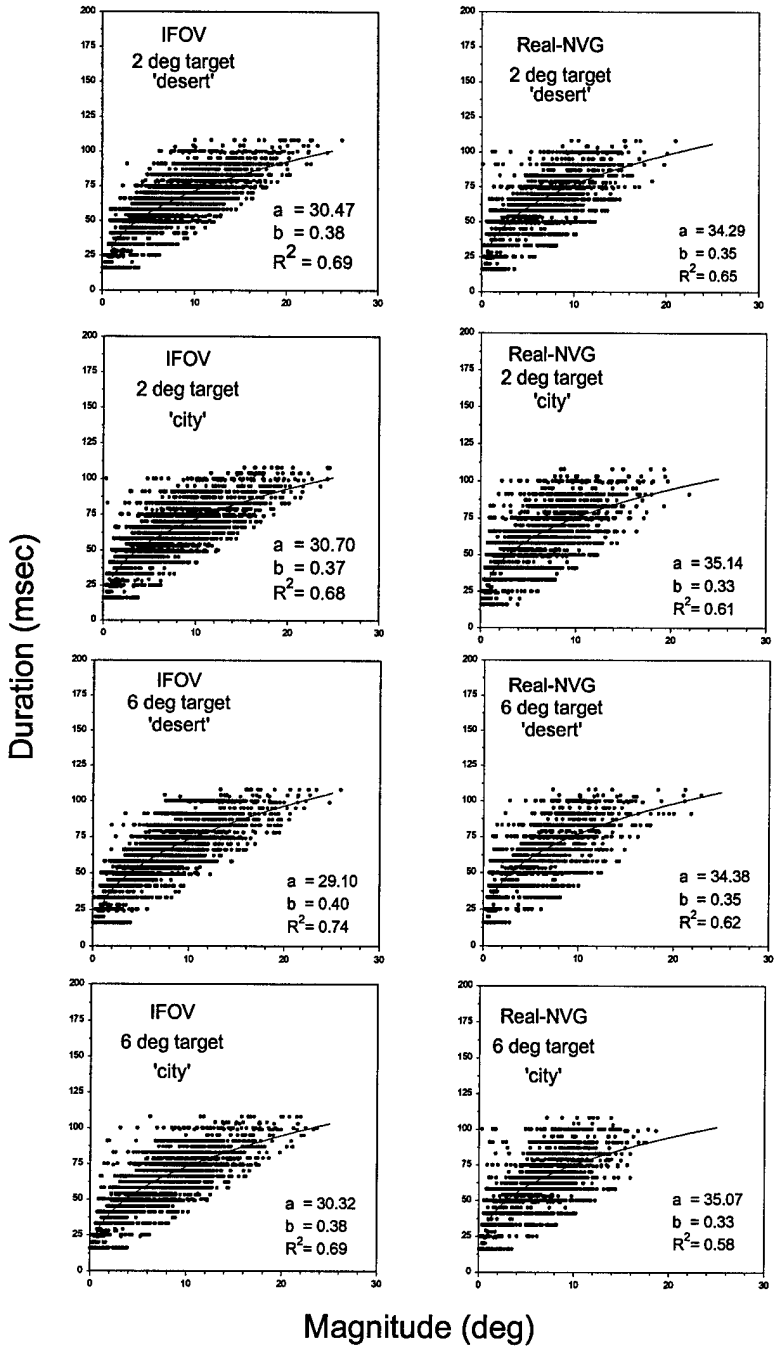


Figure 21. Main Sequence Data for the IFOV and Real-NVG Conditions

4.0 DISCUSSION

4.1 Head and Eye Movements

Head- and eye-movements (HEMs) must be properly coordinated in order to perform efficient visual search, especially over large fields-of-view. Most research on the coordination of HEMs involves controlled and limited responses wherein an observer's gaze point is moved from one point, specified by the experimenter, to another located some distance away. Under these conditions, HEMs usually display a characteristic pattern that consists of an initial eye saccade, whose amplitude is approximately equal to the target displacement, followed by a head movement, which serves to return the eye to an approximately centered position in the orbit (Bizzi, 1981; Leigh & Zee, 1991). This is a somewhat artificial situation, however, in that the observer is given relatively little choice in determining either the timing or displacement of the HEMs. Although stimulus factors are assumed to be a major determinate of the HEMs made under all viewing conditions, observers viewing or searching real-world scenes are much freer to choose both the timing and location of successive fixations (see, e.g., Andrews & Coppola, 1999). While it is recognized that there will be common features in the HEMs made in diverse viewing situations, the present study is concerned with those made by observers who have been asked to search for minimally visible targets in complex scenes.

4.2 Head and Gaze Scanpaths

When asked to view a visual scene or image, most observers display a characteristic gaze pattern known as a scanpath (cf., Yarbus, 1967). Scanpaths have been used to study visual and cognitive function (cf. Groner & Menz, 1985;, Gould, 1967; Noton & Stark, 1971; Stark & Ellis, 1981; Bellenkes et al. 1997), and as measures of skilled behavior and workload (cf., Thackray & Touchstone, 1980; Tole, Stephens, Harris, & Ephrath, 1982; Katoh, 1997). Patterns that appear similar to scanpaths have also been found when observers search for objects in the visual field. For instance, Enoch (1960) described the scanpaths used by his observers as they searched for various targets in aerial photographs. He identified general search patterns which were both idiosyncratic for a given observer and consistent over a wide variety of targets and background imagery. Enoch described these search patterns as "lateral back and forth", "up and down", "spiral", "closed square", or "non-directive." Stark, Yamashita, Tharp, and Ngo (1990) also

measured scanpaths for observers searching for targets on various background images. They described the scanpaths they found as "vertical", "horizontal", "circular", "oscillating vertical" or "oscillating horizontal", which is in general agreement with the descriptions given by Enoch (1960). The head and gaze scanpaths obtained in the present study, and shown in Figs. 2 and 3, and Appendix C, are generally similar to those reported by both Enoch (1960) and Stark et al. (1990).

For the smallest IFOV (i.e., 10°) tested in the present study, the head and gaze scanpaths were very nearly identical in extent (again, see Figs. 2 and 3, and Appendix C). For at least one condition for each observer, however, the gaze scanpath exceeded the head scanpath by 5°-10° when IFOV=20°. The difference in the extent of the two scanpaths was even larger (10°–20°) for IFOV=40°, and was evident under all testing conditions. Also, for several of the observers, the difference in the extent of the head and gaze scanpaths was asymmetrical in that it was larger for scans in one direction than for scans in the opposite direction (see, e.g., LM, 2° target, IFOV=20°).

All twelve observers tested here used a formalized scanpath at some point while performing the assigned search task, and nine of the twelve did so consistently under the various testing conditions. Whether free (i.e., random or unpredictable) or formal scanning is most appropriate may be dependent on the nature of the search task to be performed. For instance, both Ford, White, & Lichenstein. (1959) and Enoch (1960) found that when performing a free search, observers first scanned more or less randomly, presumably to survey the stimulus field, and then used a more systematic scanpath until the target was detected. Both of these studies were similar to the present study in that observers were asked to search for a target which might be located anywhere in the visual field and which might not be visible initially. Harker & Jones (1981) found, however, that helicopter pilots searching for potential terrain hazards, such as trees in or near the line of flight, tended to use irregular scanpaths. Based on these data and a review of the relevant literature, Kotulak (1992) recommends that free scanning be used by helicopter pilots performing what he calls "flight path scanning." Kotulak describes the visual task used by Harker and Jones as providing spatial uncertainty as to the target location. The visual task used in the present study involves even more spatial uncertainty, which may explain the difference in scanning behavior of the respective observers. This interpretation is also consistent with the fact

that cockpit instrument scanning involves the least spatial uncertainty and is typically performed using unsystematic scanpaths [although ordered sequences of eye movements among various instruments have been reported for this task (Tole et al., 1982)].

It should be noted that visual search is a complex activity, which was not explicitly studied here. Further, we did not provide response feedback to our observers, nor did we provide any information that may have increased the efficiency of their search strategies. Thus, our data cannot be used to determine the most efficient method of visual search using NVGs. Although forcing observers to follow a formal scanpath has not been shown to increase search efficiency (Gale & Worthington, 1983; Thackray & Touchstone, 1980; Townsend & Fry, 1960), given that NVG users apparently modify their scanning strategy depending on the task they are performing, we see no basis for concluding that a formal (patterned) scanpath is inefficient when performing free visual search such as was required of the observers in the present study.

4.2.1 Head-Scan Magnitude, Period, and Velocity

Given the large background field used in the present study, the head scanpath is a major determinant of how completely and efficiently visual search is carried out. Further, given the systematic head-scans used by the observers in the present study, it might be expected that the magnitude and velocity of the of the head-scans would be dependent on characteristics of both the background and target imagery. The data of Figure 6 show that this was not the case, however, for the magnitude of the head scanpaths, which decreased as IFOV was increased for both of the target sizes and levels of background detail tested. Analogous results were obtained for the period of the head scanpaths, which increased with IFOV but again did not vary significantly with either target size or level of background detail (Figure 8). The data of Figs. 6 and 8 are consistent in that they indicate that the observers relied less on head scanning as IFOV was increased. As shown by the data of Figure 10, however, head-scan velocity did not change significantly with IFOV. Further, head-scan velocity was affected by target size, but not by background detail. This difference would be consistent with the assumption that target size primarily affects the amount of foveal processing required in the search task, while background detail affects peripheral processing. This issue will be discussed further in the context of gaze magnitude and fixation duration (see below).

Finally, it has been reported that observers performing free visual search often do not inspect all portions of the image equally (Ford et al., 1959; Enoch, 1960; Townsend & Fry, 1960. However, the gaze (i.e., head + eye) scans of the observers in the present study covered the full horizontal and vertical extent of the background image. This may be related to the fact that the background imagery was static, familiar to the observers, and for the most part spatially homogeneous.

4.3 Gaze-Saccade Magnitude and Direction

Because of the many possible combinations of stimulus configurations and visual tasks, it is difficult to find eye movement data that can be directly compared across studies. Probably the most pertinent characteristic of the present search task, in the sense of defining or limiting the range of possible gaze saccades, is that no spatial positions were identified as to where the target stimulus might appear. In one of the earliest studies of eye movements during free search, Ford et al. (1959) asked observers to search for a dim 0.5° white spot that might be present anywhere on a 30° white background. With 5 sec search trials, they found a gaze magnitude distribution with a relatively broad peak from about 3° to 9° and a mean of 8.6°. Groner and Menz (1985) asked observers to search for a small dot pattern in a much larger random pattern of similar dots. Their display subtended 26° × 26° but was constructed so as to influence their observers to search within one of nine, 8.7° × 8.7° subareas. Ten of twelve observers showed mean saccadic amplitudes of between about 3° and 4.5°, while the other two observers showed mean amplitudes of about 5.4° and 6.8°, respectively. Finally, Andrews and Coppola (1999) measured saccadic magnitudes as observers searched for a "Waldo" icon in a 44° × 54° image. The mean saccadic magnitude for 15 observers was about 5.2°. The mean of 8.6° reported by Ford et al. (1959), for a 30° background image, is consistent with the present mean gaze magnitudes (see Table 1) of about 7° and 9° for IFOVs of 20° and 40°, respectively. Likewise, the mean gaze magnitude of about 5.5° for the 10° IFOV condition of the present study is similar to gaze magnitudes reported by Groner and Menz (1985) for an effective 8.7° FOV. The mean saccadic magnitude of 5.2° reported by Andrews and Coppola (1999) is somewhat lower than the magnitudes found by both Ford et al. (1959) and the present study for comparable FOVs. These differences may be related to the fact that locating the icon in the Andrews and Coppola study required that observers inspect the fine spatial detail of the background image.

Gaze-saccade magnitude is one of the most fundamental characteristics of the head and eye movements associated with visual search. As was the case with both head-scan magnitude and period, gaze-saccade magnitude also varied significantly as a function of IFOV (see Figs. 11 and 12, Table 1, and Appendix D). In fact, the decrease in head-scan magnitude with IFOV described earlier is effectively counteracted by an increase in gaze magnitude. This results in the same total display area being foveated in the course of the visual search. As might be expected, given the compensatory relationship between head-scan magnitude and gaze magnitude, the latter, like the former, was virtually unaffected by either target size or background detail (see Figure 12).

Image detail in the visual periphery is often used to guide HEMs (Gauthier, Semmlow, Vercher, Pedrano, & Obrecht,1991; Semmlow, Gauthier, & Vercher, 1992). In the present study, the IFOV set a practical limit on gaze-saccade magnitude in that there was no stimulus to fixate if that magnitude were larger than one-half the IFOV. When provided with the smallest (i.e., 10°) IFOV, the observers performed the present search task using gaze-saccades that averaged about 5° in magnitude. Since the aperture that defined the IFOV was 5° in radius, successive saccades were made to an image point very close to where the aperture edge was located at the end of the previous saccade. For IFOV=20°, the mean gaze magnitude increased to about 7°, suggesting that the aperture in the IFOV=10° condition was effectively limiting saccade magnitude. Finally, for the IFOV=40° condition, mean gaze magnitude increased only slightly to about 9°, suggesting that only a limited portion of the peripheral image beyond 10° (i.e., the radius of the 20° IFOV aperture) was affecting gaze magnitude.

Another interpretation is that the gaze saccades used by the present observers to perform the present search task are preset to a range of about 7°–9°. The upper end of the range would thus be manifested for IFOV=40°, since the aperture would in that case not limit the size of the gaze saccades. The lower end of the range might then be interpreted as a compromise between maximizing gaze-saccade amplitude and simultaneously avoiding the aperture edge, which might exert a masking effect. This latter interpretation is also consistent with the fact that neither target size nor background detail affected the size of gaze saccades, which, as was discussed earlier in the context of head-scan magnitude, might be related to the static, homogeneous, and familiar nature of the background imagery.

Given the search patterns most often used by the present observers, head movement was directed along either horizontal or vertical paths. Thus, it is not surprising that the distributions of gaze direction shown in Figure 15 and Appendix E are relatively narrow, since large deviations from the head scan direction would be unnatural and physically difficult to perform. The slight widening of the gaze-direction distributions may be attributed to the changes in head-scan direction associated with both the horizontal and vertical scans.

4.4 Gaze-Saccade Fixation Duration

It has been well documented that little or no visual information is obtained during the saccadic eye movements that accompany visual search (cf., Matin, 1986). A much more important concomitant of visual search are the individual fixations between saccades, during which information is presumably extracted from the visual scene. Again, the present data can only meaningfully be compared to other data obtained using a qualitatively similar search task. The studies of Ford et al. (1959), Groner and Menz (1985), and Andrews and Coppola (1999), discussed earlier, fall into this category, and the mean fixation durations reported in those studies were 0.28, 0.46 and 0.21 s, respectively. The data of Ford et al. and of Andrews and Coppola are very similar to the mean of about 0.20 s obtained over all conditions in the present study. Groner and Menz speculate that the much higher value that they found was due to intensive cognitive processing, although it is not clear from their description of their search task that such processing would be necessary.

As noted above, information is obtained from the visual image primarily during the period of fixation. Thus, it might be expected that fixation duration would be the eye movement parameter most affected by the nature of both the target stimulus and the background image. The data of Figure 18 show that there was a significant decrease in mean fixation duration with IFOV for the 2° test target but not for the 6° test target. This aspect of the data is consistent with the expectation that observers would require more time to decide whether a smaller test stimulus were present, before moving on to the next location to be searched. It is less clear, however, why more time is allotted by the observers for this purpose only for the smallest IFOV. One possibility is that information beyond 5° in the periphery is available to the observers during each fixation. That information could then be used to reduce the time required to determine whether the target was present during the succeeding fixation. Less peripheral information would be

available for IFOV=10°, and thus a longer fixation duration would be required—as the data of Figure 18 indicate. O'Regan, Levy-Shoen, & Jacobs. (1983) introduced the concept of visual span in their study of eye movements during reading. They defined the visual span as the size of the region around the fixation point within which letters could be recognized, and assumed that it was primarily determined by the physical characteristics of the stimulus (i.e., size, spacing, sharpness, etc.). Although O'Regan et al. found no relationship between the visual span and eye-movements during reading, Jacobs (1986) used a simpler search task and found that mean fixation duration decreased as the visual span was increased–a result consistent with the present data.

4.5 Main Sequence Data for Free Visual Search

The term *main sequence* was coined by Bahil, Clark, & Stark. (1975) and refers to the relationship between saccadic magnitude and either saccadic duration or peak velocity. The sampling rates we used were not sufficient to accurately estimate peak velocity, and so only magnitude-duration data are considered here.

Bahill et al. (1975) describe their log saccadic magnitude vs. log saccadic duration data as nonlinear over a three log unit range of saccadic magnitudes (from about 0.5°–50°). However, given the relatively large spread in their data for magnitudes less than about 1°, it appears that a linear function would fit the data as well as any other over the range 0.5°–10°. As their data are plotted using log-log coordinates, a linear function suggests that saccadic duration and magnitude are related by a power function. Further, Becker (1991) reviewed much of the relevant data in this area and conclude that a power function provides an acceptable fit to saccadic duration vs. saccadic magnitude data for magnitudes between about 0.5° and 5.0°. We have therefore used a power function to characterize the present data. It should also be noted that we used very liberal selection criteria to determine which gaze saccades were analyzed. That is, virtually all responses (whose duration was less than 108 ms, see Method) were included in the data analysis. Further, the present data are free responses as opposed to controlled responses to discrete and somewhat predictable stimulus locations. As a result, the variability in the present data is generally greater than that reported in some of the other studies referenced here.

After reviewing much of the literature on saccadic eye movements, Becker (1991) has

concluded that a power function of the form duration = a · (magnitude)^b can provide an adequate fit to duration-magnitude main sequence data for saccadic durations from about 0.5°–5.0°. He contends that the exponent, b, for a fit over this data range is about 0.15–0.20. Becker (1991) also plots data from other studies and shows that they are well fit by a linear function for saccadic magnitudes from about 5° to 50°. He claims that there are significant deviations from the linear function for saccadic magnitudes below 5°, but these deviations are not evident in the plotted data. Baloh, Sills, Kumley, & Honrubia. (1975) present data that suggest that duration vs. amplitude functions are linear over a magnitude range of 6°–90°, although they note that some observers' data show a "mild curvature". Finally, Van Gisbergen, Van Opstal, & Schoenmakers. (1985) have concluded that a linear function is sufficient for fitting duration vs. magnitude data over a magnitude range of about 5°–30°.

Based on his analysis as described above, Becker (1991) seems to be suggesting that main sequence duration-magnitude data be fitted in a piecewise fashion using separate power functions (note that a linear function is a power function with an exponent of one). The data of this study extend over a saccadic magnitude range of about 0.5°–20°, and we see no theoretical or practical reason to fit them with more than one function. Therefore, we have fitted a two-parameter power function of the form described above. The results are shown for each testing condition in the individual panels of Figs. 19 and 20. The exponents obtained for the present data range from 0.24–0.38 and hence are much larger than those described by Becker (1991). Thus, main sequence relations may be dependent on whether a well-defined target is present, as was the case for the data plotted by Becker (1991), but was not the case in the present effort. Further, a larger exponent is associated with a less compressive function, i.e., a function by which duration increases relatively more as magnitude increases. We might, therefore, speculate in the context of models of saccadic generation (Becker, 1991) that the lack of a definitive target in the present study increases the saccadic error signal that results from either visual or non-visual (Jürgens, Becker, & Kornhuber, 1981) discrepancies between the target and eye position. A larger error signal would be expected to result in larger and more variable saccades, which would be consistent with the larger power function exponents found here. Of course, we cannot exclude the possibility that the error signal is qualitatively different in free search as compared with the situation when a well-defined target is available. In the latter case, the relationship between

saccadic duration and magnitude may be determined, at least partially, by higher-level (even cognitive) processes that determine the appropriate saccadic characteristics for a given set of stimulus conditions.

Main sequence data have been used to quantify differences in the dynamics of the eye movements associated with various visual tasks (Eizenman, Frecker, & Hallett, 1984; Epelboim, Steinman, Kowler, Pizlo, Erkelens, & Collewijn, 1997; Smit, Van Gisbergan, & Cools, 1987), as well as differences for various stimulus conditions for the same task (Hooge & Erkelens, 1999). Our tentative conclusion is that, under the present study conditions, the form of the function relating saccadic duration to saccadic magnitude does not change significantly with IFOV, target size, or level of background detail. This conclusion must be qualified somewhat, however, because of the large variability in the data both among observers and among conditions for a given observer. This variability could be reduced by selecting and analyzing saccades that occurred over restricted portions of the gaze scan. For instance, there were typically between four and eight gaze saccades during each of the horizontal scans used by our observers during the present search task (see Figure 3). It is likely that response variability would be reduced if some subset of these saccades were analyzed in isolation.

4.6 Implications for the Use of NVGs

For the restricted category of search required of the present observers, and given the systematic head scanning used by those observers, it would appear that there is no advantage to using an IFOV larger than about 15°–20° (see Section 4.3). There may, in fact, be an advantage in the smaller IFOV if it helps the observer maintain the chosen head-scan pattern. The disadvantage of a smaller IFOV, of course, is that the observer would be less likely to detect unexpected changes in the visual field (e.g., changes in peripheral targets, or the appearance of or change in extraneous targets that may nevertheless be relevant to the search process), which are likely to be important when search is performed in the real-world. A larger IFOV is also to be desired if the NVG user might be required to perform other flight-related tasks in addition to a free search for near-threshold targets.

There is apparently no consensus in the literature as to whether visual search performance can be improved by training observers to use specific, formalized scanning patterns. Several

early studies concluded that observers tended to use scan patterns that did not adequately cover the area being searched (Enoch, 1960; Ford et al., 1959). Clearly, formalized scanning would solve this problem, but it does not necessarily improve search performance (Gale & Worthington, 1983; Thackray & Touchstone, 1980). Banks et al. (1971) also noted that many observers do not scan efficiently, and further suggest that operators using night-vision devices be trained in specific search techniques that include using rectangular search patterns and a variable scanning rate. The utility of head movement training in this context has also been investigated (Seagull & Gopher, 1997). The observers tested in the present IFOV condition were not trained in any way and yet used generally consistent scan patterns under diverse viewing conditions. This would suggest that formal training is not necessary unless the scanning behavior chosen is shown to be inefficient. Although more study is required on this issue, we are aware of no experimental evidence of such inefficiency.

We found no significant differences in the head and eye movements used by naïve observers and those who have had experience using NVGs. It should be noted, however, that the task that the observers were asked to perform in the present study is in no way specific to NVG viewing and is not necessarily practiced more by NVG users. There is relatively little formal training of scanning strategies for NVG users, and even if there were, it would most likely serve to improve context-specific tasks such as the identification of familiar targets or the distinguishing of targets from known imaging artifacts unique to NVGs. To determine the search strategies most appropriate when NVGs are used, tasks specific to NVG use must be tested. In addition, if learning rates are an issue, both practiced and naïve observers should be tested both on tasks that both groups are familiar with and on novel tasks that will allow generalization and transfer of training to be assessed.

5.0 REFERENCES

Andrews, T.J. & Coppola, D.M. (1999). Idiosyncratic characteristics of saccadic eye movements when viewing different visual environments. *Vision Research*, **39**, 2947-2953.

Bahill, A.T., Clark, M.R., & Stark, L. (1975). The main sequence, a tool for studying human eye movements. *Mathematical Biosciences*, **24**, 191-204.

Baloh, R.W., Sills, A.W., Kumley, W.E., & Honrubia, V. (1975). Quantitative measurement of saccade amplitude, duration, and velocity. *Neurology*, **25**, 1065-1070.

Banks, J.H., Sternberg, J.J., Cohen, B.J., & DeBow, H. (1971). Improved search techniques with passive night vision devices. *Technical Research Report 1169*, U.S. Army Behavior and Systems Research Laboratory, Arlington, VA.

Becker, W. (1991). Saccades. In: *Vision and Visual Dysfunction, Vol. 8. Eye Movements* (R.H.S. Carpenter, Ed.). pp. 95-137. London: MacMillan

Bellenkes, A. H., Wickens, C. D., Kramer, A. F..(1997) Visual scanning and pilot expertise: The role of attentional flexibility and mental model development In *Aviation, Space and Environmental Medicine,* 68.

Bizzi, E. (1981). Eye-head coordination. In: *Handbook of Physiology, Section 1, Volume 2, Part 2,* (J.M. Brookhart & V.B. Mountcastle, Eds.),. Bethesda: American Physiological Society.

Eizenman, M., Frecker, R.C., & Hallett, P.E. (1984). Precise noncontacting measurement using the corneal reflex. *Vision Research*, **24**, 167-174.

Enoch (1960). Natural tendencies in visual search of a complex display. In: *Visual Search Techniques* (A. Morris & E.P. Horne, Eds.). pp. 187-193, National Research Council Publication 172, Washington, D.C.

Epelboim, J., Steinman, R.M., Kowler, E., Pizlo, Z., Erkelens, C., & Collewijn, H. (1997). Gaze-shift dynamics in two kinds of sequential looking tasks. *Vision Research*, **37**, 2597-2607.

Ford, A., White, C.T., & Lichtenstein, M. (1959). Analysis of eye movements during free search. *Journal of the Optical Society of America*, **49**, 287-292.

Gale, A.G. & Worthington, B.S. (1983). The utility of scanning strategies in radiology. In: *Eye Movements and Psychological Functions: International Views* (R. Groner, C. Menz, D.E. Fischer, & R.A. Monty, Eds.). New York: Lawrence Erlbaum.

Gauthier, G.M., Semmlow, J.L., Vercher, J.-L., Pedrano, C., & Obrecht, G. (1991). Adaptation of eye and head movements to reduced peripheral vision. In: *Oculomotor Control and Cognitive Processes* (R. Schmid & D. Zambarbieri, Eds.). North-Holland: Elsevier.

Gould, J.D. (1967). Pattern recognition and eye-movement parameters. *Perception & Psychophysics*, **2**, 399-407.

Gould, J.D. (1973). Eye movements during visual search and memory search. *Journal of Experimental Psychology*, **98**, 184-195.

Groner, R. & Menz, C. (1985). The effect of stimulus characteristics, task requirements and individual differences on scanning patterns. In: *Eye Movements and Human Information Processing* (R. Groner, G.W. McConkie, & C. Menz, Eds.). North-Holland: Elsevier.

Harker, G.S. & Jones, P.D. (1981). Analysis of pilots' eye movements during helicopter flight. *Technical Report PMT-ET-0015-81*, Naval Training Center, Orlando, Florida.

Hooge, I.T.C. & Erkelens, C.J. (1999). Peripheral vision and oculomotor control during visual search. *Vision Research*, **39**, 1567-1575.

Jacobs, A. M. (1986). Eye-movement control in visual search: How direct is visual span control. *Perception & Psychophysics*, **39**, 47-58.

Jürgens, R., Becker, W., & Kornhuber, H.H. (1981). Natural and drug-induced variations of velocity and duration of human saccadic eye movements: Evidence for a control of the neural pulse generator by local feedback. *Biological Cybernetics*, **39**, 87-96.

Katoh, Z (1997). Saccade amplitude as discriminator of flight types, *Aviation, Space, and Environmental Medicine*, 68.

Kotulak, J.C. (1992). Methods of visual scanning with night vision goggles. *USAARL Report 92-10*, U.S. Army Aeromedical Research Laboratory, Fort Rucker, Alabama.

Leigh, R. J. & Zee, D. S. (1991). *The Neurology of Eye Movements, Edition 2.* (Ch. 7, Head and eye movements.) Philadelphia: F.A. Davis Co.

Matin, L. (1986). Visual Localization and eye movements. In: *Handbook of Perception and Human Performance, Vol.1* (K.R. Boff, L. Kaufman, & J.P. Thomas, Eds.), New York: John Wiley.

Noton, D. & Stark, L. (1971). Scanpaths in eye movements during pattern perception. *Science*, **171**, 308-311.

O'Regan, J.K., Lévy-Schoen, A., & Jacobs, A.M. (1983). The effect of visibility on eye movement parameters in reading. *Perception & Psychophysics*, **34**, 457-464.

Seagull, F.J. & Gopher, D. (1997). Training head movement in visual scanning: An embedded approach to the development of piloting skills with helmet-mounted displays. *Journal of Experimental Psychology*, Applied, **3**, 163-180

Semmlow, J.L., Gauthier, G.M., & Vercher, J.-L. (1992). Identification of peripheral visual images in a laterally restricted gaze field. In: *The Head-Neck Sensory Motor System* (A. Berthoz, W. Graf, & P. P. Vidal, Eds.), New York: Oxford University Press

Smit, A.C., Van Gisbergen, J.A.M., & Cools, A.R. (1987). A parametric analysis of human saccades in different experimental paradigms. *Vision Research*, **27**, 1745-1762.

Stark, L. & Ellis, S.R. (1981). Scanpaths revisited: Cognitive models direct active looking. In: *Eye Movements: Cognition and Visual Perception* (D.F. Fisher, R.A. Monty, & J.W. Senders, Eds.), pp. 193-226. Hillsdale, N.J.: Lawrence Erlbaum.

Stark, L., Yamashita, I., Tharp, G., & Ngo, H.X. (1990). Keynote Lecture: Search patterns and search paths in human visual search. In: *Visual Search 2* (D. Brogan, A. Gale, & K. Carr, Eds.), London: Taylor and Francis.

Thackray, R.I. & Touchstone, R.M. (1980). Visual search performance during simulated radar observations with and without a sweepline. *Aviation, Space, and Environmental Medicine,* **51**, 361-366.

Tole, J.R., Stephens, A.T., Harris, R.L. Sr., & Ephrath, A.R. (1982). Visual scanning behavior and mental workload in aircraft pilots. *Aviation, Space, and Environmental Medicine,* **53**, 54-61.

Townsend, C.A. & Fry, G.A. (1960). Automatic scanning of aerial photographs. In: *Visual*

Search Techniques (A. Morris & E.P. Horne, Eds.). pp. 194-210. National Research Council Publication 172, Washington, D.C.

Van Gisbergen, J.A.M., van Opstal, A.J., & Schoenmakers, J.J.M. (1985). Experimental test of two models for the generation of oblique saccades. *Experimental Brain Research*, **57**, 321-336.

Yarbus. A.L. (1967). *Eye Movements and Vision*. New York: Plenum Press.

6.0 APPENDICES

6.1 Appendix A

Source code for the program used to present the background imagery and test targets used in the present study. The program includes provisions for collecting and analyzing target detection data although such data were not required in the present study.

```
/*
 *                  Program:            test_sig.c
 *                  Created:            15-FEB-95
 *                  Created By:         Craig A. Vrana
 *                  Last Modified: 07-MAY-96
 */
/*
       To use this program, you must know 3 things:
       1)      The distance from the screen to the subjects eyes.
       2)      The size of the 1024x1024 image in meters as projected on the screen.
       3)      The name and location of a 1024x1024 flat binary image used as
               background to the targets.

       The size of the image in meters is entered below as the constant:
       IMAGE_METERS
*/
#include       <get.h>
#include <gl.h>
#include <device.h>
#include <stdio.h>
#include       <math.h>
#include       <limits.h>
#include <unistd.h>
#include <sys/schedctl.h>
#include <sys/syssgi.h>
#include <sys/types.h>
#include <sys/times.h>
#include       <sys/time.h>
#define TARGETS                         1
#define ON                                      1
#define OFF                             0
#define UP                                      0
#define DOWN                            1
#define IMAGE_SIZE                      1024
#define BACKGROUND_A            "dsrt1_lc.img\0"
#define BACKGROUND_B            "new_city_lc.img\0"
#define TARGET_SIZE                     33              /* Use odd numbers to balance target */
#define TARGET_SIZE_A           33
#define TARGET_SIZE_B           97
#define min(a,b)                        (((a) < (b)) ? (a) : (b))
#define max(a,b)                        (((a) > (b)) ? (a) : (b))
#define PROJECTION_COLOR        BLUE
#define SHORT                           1               /* For short bell.              */
#define X_IMAGE_START           128     /* Distance from left side that image starts */
#define X_IMAGE_END                     1151    /* Distance from left side that image ends   */
#define X_CENTER                        ((int)(XMAXSCREEN / 2.0 + 0.5))
#define Y_CENTER                        ((int)(YMAXSCREEN / 2.0 + 0.5))
#define RADIUS                          15              /* Radius of cross-hair targets      */
/* The following constant must be changed, if the image size changes.           */
#define IMAGE_METERS            1.835   /* Length and height of image in meters      */
#define PIXELS_PER_METER        (IMAGE_SIZE / IMAGE_METERS)
#define STEPS                           4091 /* approx 10 secs */
#define TRIALS                          10
#define CYCLES_PER_TARGET       8
#define PPC                             (TARGET_SIZE / CYCLES_PER_TARGET)
#define SIGNAL_COLOR            512
#define BLIP_SIZE                       30
#define START_DELAY                     2               /* Length of time before test starts.
             */
#define PREVIEW_TIME            4               /* Number of seconds target is previewed
       */
#define CROSSHAIR_TIME          2               /* Seconds crosshair is displayed before trial
       */
#define N                                       (TARGET_SIZE)
#define D                                       16.0                                    /* 32.0
       */
#define F                                       5.00                                    /*
Cycles per target       */
#define Theta                           (M_PI)                          /* Orientation of
target  */
#define PHx                             (M_PI/(2.0*F))          /* M_PI/4.0     */
```

```
#define PHy                                    (0.0)                    /* M_PI/4.0
        */
/*  Global Variables  */
FILE    *outfile;
char  infile[64];
char  targfile[64];
char  datafile[64];
float   distance = 1.5, /*     Distance from screen to eyes in meters       */
                Fx, Fy;
int     locs = 0;
static int      target_locs[2][64];    /* Hard coded values into this array due to stability
problem */
short   image_array[IMAGE_SIZE][IMAGE_SIZE];
short target_array1[TARGET_SIZE][TARGET_SIZE];
/*
short target_array2[TARGET_SIZE][TARGET_SIZE];
short target_array3[TARGET_SIZE][TARGET_SIZE];
short target_array4[TARGET_SIZE][TARGET_SIZE];
short target_array5[TARGET_SIZE][TARGET_SIZE];
*/
short source_array1[TARGET_SIZE][TARGET_SIZE];
/*
short source_array2[TARGET_SIZE][TARGET_SIZE];
short source_array3[TARGET_SIZE][TARGET_SIZE];
short source_array4[TARGET_SIZE][TARGET_SIZE];
short source_array5[TARGET_SIZE][TARGET_SIZE];
*/
float step_array1[TARGET_SIZE][TARGET_SIZE];
/*
float step_array2[TARGET_SIZE][TARGET_SIZE];
float step_array3[TARGET_SIZE][TARGET_SIZE];
float step_array4[TARGET_SIZE][TARGET_SIZE];
float step_array5[TARGET_SIZE][TARGET_SIZE];
*/
float current_array1[TARGET_SIZE][TARGET_SIZE];
/*
float current_array2[TARGET_SIZE][TARGET_SIZE];
float current_array3[TARGET_SIZE][TARGET_SIZE];
float current_array4[TARGET_SIZE][TARGET_SIZE];
float current_array5[TARGET_SIZE][TARGET_SIZE];
*/
short show_array1[TARGET_SIZE][TARGET_SIZE];
/*
short show_array2[TARGET_SIZE][TARGET_SIZE];
short show_array3[TARGET_SIZE][TARGET_SIZE];
short show_array4[TARGET_SIZE][TARGET_SIZE];
short show_array5[TARGET_SIZE][TARGET_SIZE];
*/
static short   blip_array[BLIP_SIZE][BLIP_SIZE];
static short          *red, *green, *blue;
/*-------------------------------------------------------------------------*/
                                /*********
                                *  main  *
                                *********/
main(argc,argv)
        int argc;
        char *argv[];
{
        int     i, j,
                        loop = 0,
                        exit_value = 0,
                        image = 0;
   if(argc == 4)
   {
                strcpy(infile, argv[1]);
                if (strchr(infile, '.')==NULL)
                {
                        strcpy(targfile, infile);
                        strcat(targfile, ".loc\0");
                        strcat(infile, ".img\0");
                }
                else
```

```
                {
                        strcpy(targfile, infile);
                        strcpy(strrchr(targfile, '.'), "\0");
                        strcat(targfile, ".loc\0");
                }
                distance = atof(argv[2]);
                strcpy(datafile, argv[3]);
                if (strchr(datafile, '.')==NULL)
                {
                        strcat(datafile, ".dat");
                }
        }
        else
    {
                /*      get_distance(); Uncomment this call, to ask for screen distance.    */
                get_data();
                get_image();
    }
        get_info();
        printf("Distance = %f meters\n", distance);
        target_locs[0][0] = 100;
        target_locs[1][0] = 912;
        target_locs[0][1] = 200;
        target_locs[1][1] = 826;
        target_locs[0][2] = 376;
        target_locs[1][2] = 925;
        target_locs[0][3] = 438;
        target_locs[1][3] = 864;
        target_locs[0][4] = 580;
        target_locs[1][4] = 794;
        target_locs[0][5] = 698;
        target_locs[1][5] = 880;
        target_locs[0][6] = 815;
        target_locs[1][6] = 702;
        target_locs[0][7] = 694;
        target_locs[1][7] = 645;
        target_locs[0][8] = 460;
        target_locs[1][8] = 625;
        target_locs[0][9] = 287;
        target_locs[1][9] = 696;
        target_locs[0][10] = 159;
        target_locs[1][10] = 677;
        target_locs[0][11] = 106;
        target_locs[1][11] = 611;
        target_locs[0][12] = 85;
        target_locs[1][12] = 504;
        target_locs[0][13] = 126;
        target_locs[1][13] = 366;
        target_locs[0][14] = 204;
        target_locs[1][14] = 314;
        target_locs[0][15] = 474;
        target_locs[1][15] = 465;
        target_locs[0][16] = 515;
        target_locs[1][16] = 324;
        target_locs[0][17] = 397;
        target_locs[1][17] = 142;
        target_locs[0][18] = 800;
        target_locs[1][18] = 509;
        target_locs[0][19] = 755;
        target_locs[1][19] = 413;
        target_locs[0][20] = 840;
        target_locs[1][20] = 349;
        target_locs[0][21] = 744;
        target_locs[1][21] = 251;
        target_locs[0][22] = 736;
        target_locs[1][22] = 156;
        target_locs[0][23] = 139;
        target_locs[1][23] = 114;
        target_locs[0][24] = 556;
        target_locs[1][24] = 233;
        locs = 25;
/*      locs = get_locations();       */
```

```
        printf("Number of target locations = %d\n", locs);
        randomize_locations(locs);
        if(!strcmp(infile, BACKGROUND_A))
        {
                image = 1;
        }
        else if(!strcmp(infile, BACKGROUND_B))
        {
                image = 2;
        }
        else
        {
                printf("Enter either new_city_lc or dsrt1_lc as the starting background.\n");
                exit(-1);
        }
        loop = 2;
        /* system("ifconfig 'ec0' down");     /* Shut down the network */
        blanktime(0);
        graph_init();
        color(0+256);
        clear();                                                /* Clear       Targe   Buffer
                                        */
        swapbuffers();                          /* Display Target Buffer
                */
        clear();                                                /* Clear Fixation Buffer
                                */
        swapbuffers();                          /* Display Fixation Buffer
                */
        while(loop>0)
        {
                display_image(image);  /* Load background into Target Buffer        */
                swapbuffers();                          /* Display Target Buffer
                        */
                display_image(image);  /* Load background into Fixation Buffer      */
                display_targets();              /* Write cross-hairs to Fixation Buffer      */
                swapbuffers();                          /* Display Fixation Buffer
                        */
                while((!getbutton(MIDDLEMOUSE)) && (!getbutton(ESCKEY)));
                while(qtest())
                {
                        qreset();
                }
                swapbuffers();                          /* Display Target Buffer
                        */
                color(0+256);
                clear();                                                /* Clear Fixation Buffer
                                        */
                display_image(image);  /* Load background into Fixation Buffer      */
                display_target();                       /* Write cross-hair to Fixation Buffer
        */
                swapbuffers();                          /* Display Fixation Buffer
                        */
                while(qtest())
                {
                        qreset();
                }
                display_image(image);  /* Load background into Target Buffer        */
                /*      get_start(); */
                /*      schedctl(NDPRI, NDPHIMAX);                      /* Set process as max-
priveleged, non-degrading */
                backbuffer(FALSE);
                frontbuffer(TRUE);
                exit_value = get_values();      /*      Fixation Buffer is front buffer      */
                fflush(stdout);
                backbuffer(TRUE);
                frontbuffer(FALSE);
                --loop;
                image = 2/image;
                randomize_locations(TRIALS);
        }
        fflush(outfile);
        fclose(outfile);
```

```
        fflush(stdin);
        swapbuffers();
        restore_color_map();
        /* system("ifconfig 'ec0' up");              /* Bring network back up */
        blanktime(36000);
        gexit();

}
/*---------------------------------------------------------------------------*/
                                /*********
                                *  blip  *
                                *********/
blip(int mode)
{
        frontbuffer(TRUE);

        if(mode == 1)
        {
                mapcolor(SIGNAL_COLOR, 255, 0, 255);
                gflush();
                sginap(2);
        /*      printf("\nGot into blip on.\n"); */
        }
        else
        {
                mapcolor(SIGNAL_COLOR, 0, 0, 0);
                gflush();
                sginap(2);
/*              printf("Got into blip off.\n"); */
        }
}
/*---------------------------------------------------------------------------*/
                                /******************
                                *  display_image  *
                                ******************/
display_image(int background)
{
        int i,j;
        short in;
        FILE *fp;
        if(background == 1)
        {
                strcpy(infile,BACKGROUND_A);
                strcpy(targfile, infile);
                strcpy(strrchr(targfile, '.'), "\0");
                strcat(targfile, ".loc\0");
                printf("Location file string = %s\n", targfile);
        }
        else if(background == 2)
        {
                strcpy(infile,BACKGROUND_B);
                strcpy(targfile, infile);
                strcpy(strrchr(targfile, '.'), "\0");
                strcat(targfile, ".loc\0");
                printf("Location file string = %s\n", targfile);
        }
        else
        {
                printf("Incorrect background specified\n");
                exit(-1);
        }

        fp = fopen(infile,"r");
        if(fp == NULL)
        {
                printf("Could not open image file %s\n\n", infile);
                exit(-1);
        }

        for(j=1023; j>(-1); j--)
        for(i=0; i<1024; i++)
        {
```

```
                image_array[j][i] = getc(fp)+256;
        }
        fclose(fp);
        rectwrite(128,0,1151,1023,image_array);
}
/*-------------------------------------------------------------------------*/
                        /*******************
                        *  display_target  *
                        *******************/
display_target()
{
        /*      Center Target  */
        draw_target(X_CENTER, Y_CENTER, RADIUS);
}

/*-------------------------------------------------------------------------*/
                        /********************
                        *  display_targets  *
                        ********************/
display_targets()
{
        /*      Center Target  */
        draw_target(X_CENTER, Y_CENTER, RADIUS);
        /* 5 degree targets    */
        draw4targets(5.0);
        /* 10 degree targets   */
        draw4targets(10.0);
        /* 20 degree targets   */
        draw4targets(20.0);
        /* 31.43 degree targets       */
        draw4targets(31.43);
        /* 37.5 degree targets */
        draw4targets(37.5);
}
/*-------------------------------------------------------------------------*/
                        /****************
                        *  draw_target  *
                        ****************/
draw_target(x, y, radius)
        int     x,
                        y,
                        radius;
{
        int     vector[2];
        color(WHITE);
        circ(x, y, radius);
        bgnline();
                color(BLACK);
                vector[0] = x;
                vector[1] = y - radius;
                v2i(vector);
                vector[0] = x;
                vector[1] = y + radius;
                v2i(vector);
        endline();
        bgnline();
                color(BLACK);
                vector[0] = x - radius;
                vector[1] = y;
                v2i(vector);
                vector[0] = x + radius;
                vector[1] = y;
                v2i(vector);
        endline();
}
/*-------------------------------------------------------------------------*/
                        /****************
                        *  draw4targets  *
                        ****************/
draw4targets(degree_offset)
        float   degree_offset;
{
```

```
        int     pixel_offset;
        float   meter_offset;
        meter_offset = distance * tanf(degree_offset*M_PI/180.0);
        pixel_offset = (int)(meter_offset * PIXELS_PER_METER + 0.5);
        draw_target(X_CENTER+pixel_offset, Y_CENTER, RADIUS);
        draw_target(X_CENTER-pixel_offset, Y_CENTER, RADIUS);
        draw_target(X_CENTER, Y_CENTER+pixel_offset, RADIUS);
        draw_target(X_CENTER, Y_CENTER-pixel_offset, RADIUS);
}
/*-------------------------------------------------------------------------*/
                /**************
                *  get_data  *
                **************/
get_data()
{
        printf("\nEnter file name to store data : ");
        scanf("%s", datafile);
        printf("\n\n");
        if (strchr(datafile, '.')==NULL)
        {
                strcat(datafile, ".dat");
        }
}
/*-------------------------------------------------------------------------*/
                /*****************
                *  get_distance  *
                *****************/
get_distance()
{
        printf("\nEnter the viewing distance from the screen in meters : ");
        scanf("%f", &distance);
        printf("\n\n");
}               /*** get_distance ***/
/*-------------------------------------------------------------------------*/
                /**************
                *  get_image  *
                **************/
get_image()
{
        printf("\nEnter the name of the first background image : ");
        scanf("%s", infile);
        printf("\n\n");
        if (strchr(infile, '.')==NULL)
        {
                strcpy(targfile, infile);
                strcat(targfile, ".loc\0");
                strcat(infile, ".img\0");
        }
        else
        {
                strcpy(targfile, infile);
                strcpy(strrchr(targfile, '.'), "\0");
                strcat(targfile, ".loc\0");
        }
}
/*-------------------------------------------------------------------------*/
                /*************
                *  get_info  *
                *************/
get_info()
{
        char            text[256];
        if ((outfile = fopen(datafile, "w")) == NULL)
        {
                printf("Error opening output file\n");
                exit(-1);
        }

        fprintf(outfile, "#First background image: %s\n", infile);
        fflush(outfile);
        printf("Enter first stimulus size (32 or 96):\t");
        scanf("%s", text);
```

```
        printf("\n\n");
        fprintf(outfile, "#First stimulus size: %s\n", text);
        fflush(outfile);

        strcpy(text, "\0");
        fflush(stdin);

        printf("Enter a comment :\t");
        gets(text);
        printf("\n\n");
        fprintf(outfile, "#Comment: %s\n", text);
        fflush(outfile);
}
/*-------------------------------------------------------------------------*/
                                /******************
                                *  get_locations  *
                                ******************/

int get_locations()
{
        int     x,
                        y,
                        locations = 0;
        FILE *fp;
        printf("\ntargfile = %s\n", targfile);
        if(1 + access(targfile, F_OK | R_OK))
        {
                        printf("%s exists and is readable.\n", targfile);
        }
        else
        {
                return(0);
        }

        fp = fopen(targfile,"r");
        if(fp == NULL)
        {
                printf("Could not open target location file %s\n", targfile);
                printf("Using random target locations.\n\n");
                return(0);
        }
        while(fscanf(fp, "%d %d\n", &x, &y) != EOF)
        {
                target_locs[0][locations] = x;
                target_locs[1][locations] = y;
                locations++;
        }
        fclose(targfile);
        return(locations);
}
/*-------------------------------------------------------------------------*/
                                /**************
                                *  get_start  *
                                **************/

get_start()
{
        short response;
        long    value = 0;
        int i;
        qreset();
        setbell(SHORT);
        ringbell();
        sginap(40);
        ringbell();
        while(!qtest());
        value = qread(&response);
        qreset();
        do
        {
                while(!qtest());
                value = qread(&response);
                qreset();
        }while(value != MIDDLEMOUSE);
```

```
        qreset();
        setbell(SHORT);
}                               /*** get_start ***/
/*-------------------------------------------------------------------------*/
                                /*********************
                                *  get_start_no_bell  *
                                *********************/
get_start_no_bell()
{
        short response;
        long    value = 0;
        qreset();
        while(!qtest());
        value = qread(&response);
        qreset();
        do
        {
                while(!qtest());
                value = qread(&response);
                qreset();
        }while(value != MIDDLEMOUSE);
        qreset();
}                               /*** get_start_no_bell ***/
/*-------------------------------------------------------------------------*/
                                /***************
                                *  get_values  *
                                ***************/
get_values()
{
        clock_t start_time,
                                end_time,
                                diff_time;
        struct  timeval         tp;
        long            q_value = 0,
                                x1,
                                x2,
                                x3,
                                x4,
                                x5,
                                y1,
                                y2,
                                y3,
                                y4,
                                y5;
        int             dec,
                                exit_flag = 0,
                                stop_flag = 0,
                                i,j,s,
                                random_value,
                                sign,
                                toggle = 0,
                                trials = 0;
        long            start,
                                end,
                                elapsed,
                                seed_value,
                                ticks_left;
        short           pixel[3],
                                response;
        /* Fixation Buffer is currently the frontbuffer      */
        readsource(SRC_FRONT);
        init_blip();
        seed_value = (long) time(0);   /* Use time to get a seed value */
        srand(seed_value);                              /* Seed pseudo-random generator */

        while(qtest())
        {
                qreset();
        }
        ticks_left = sginap(2 * CLK_TCK);     /* Wait for N seconds         */

        while(qtest())
```

```
        {
                qreset();
        }
        swapbuffers();                                          /* Show target buffer
        */
        preview_target();                                       /* Show preview of
target  */
        swapbuffers();                                          /* Show fixation
buffer          */
        while(qtest())
                qreset();
        while(trials<TRIALS)
        {
                if(qtest())
                {
                        q_value = qread(&response);
                        switch(q_value)
                        {
                                case LEFTMOUSE:
                                        break;
                                case MIDDLEMOUSE:
                                        break;
                                case RIGHTMOUSE:
                                        break;
                                case ESCKEY:
                                        fclose(outfile);
                                        return(-1);
                                        break;
                                default:
                                        printf("Error: received unknown value in queue.\n");
                                        break;
                        }
                        qreset();
                }
                if(trials<TRIALS)
                {
                        if(locs == 0)
                        {
                                printf("Number of locations = 0. Generating ...\n");
                                x1 = (rand()%(1024-TARGET_SIZE)) + 128;
                                x2 = (rand()%(1024-TARGET_SIZE)) + 128;
                                x3 = (rand()%(1024-TARGET_SIZE)) + 128;
                                x4 = (rand()%(1024-TARGET_SIZE)) + 128;
                                x5 = (rand()%(1024-TARGET_SIZE)) + 128;
                                y1 = rand()%(1024-TARGET_SIZE);
                                y2 = rand()%(1024-TARGET_SIZE);
                                y3 = rand()%(1024-TARGET_SIZE);
                                y4 = rand()%(1024-TARGET_SIZE);
                                y5 = rand()%(1024-TARGET_SIZE);
                        }
                        else
                        {
                                x1 = target_locs[0][trials] + 128;
                                y1 = target_locs[1][trials];
                        }
/*                      fprintf(outfile, "%d %d %d %d %d %d %d %d %d %d\n", x1, y1, x2, y2,
                                                                x3, y3, x4, y4, x5, y5);
*/
                        do
                        {
                                while(!qtest());                        /* Wait to start trial
        */
                                q_value = qread(&response);
                        }while(response != DOWN);
                        qreset();

                        switch(q_value)
                        {
                                case LEFTMOUSE:
                                        break;
                                case MIDDLEMOUSE:
                                        break;
```

```
                case RIGHTMOUSE:
                        break;
                case SPACEKEY:
                        break;
                case ESCKEY:
                        fclose(outfile);
                        exit(-1);
                        break;
                default:
                        printf("Error: received unknown value in queue.\n");
                        break;
        }
        qreset();
        /*      Start Trial     */
        swapbuffers();          /* Display Target Buffer      */

        blip(ON);
        blip(OFF);
        /* Use the following code for random start time from .5 to 5 secs.
        ticks_left = (long)((rand()%10) * 0.5 * CLK_TCK);
        ticks_left = sginap(ticks_left);
        */
        /* The following waits for about 15 seconds before starting the ramp */
        ticks_left = (long)(15 * CLK_TCK);
        ticks_left = sginap(ticks_left);
        if(gettimeofday(&tp,0))
        {
                printf("Start time call failed!\n\n");
        }
        start = tp.tv_sec;
        while(qtest())
        {
                qreset();
        }
        rectread(x1,y1,(x1-1+TARGET_SIZE),(y1-1+TARGET_SIZE),source_array1);

        /*
        rectread(x2,y2,(x2-1+TARGET_SIZE),(y2-1+TARGET_SIZE),source_array2);
        rectread(x3,y3,(x3-1+TARGET_SIZE),(y3-1+TARGET_SIZE),source_array3);
        rectread(x4,y4,(x4-1+TARGET_SIZE),(y4-1+TARGET_SIZE),source_array4);
        rectread(x5,y5,(x5-1+TARGET_SIZE),(y5-1+TARGET_SIZE),source_array5);
        */
        init_current_array();
        init_target_array();
        for(i=0; i<TARGET_SIZE;i++)
        for(j=0; j<TARGET_SIZE; j++)
        {
                step_array1[i][j] = ((float)target_array1[i][j] -
(float)source_array1[i][j])
                                                                /(float)STEPS;

                /*
                step_array2[i][j] = ((float)target_array2[i][j] -
(float)source_array2[i][j])
                                                                /(float)STEPS;
                step_array3[i][j] = ((float)target_array3[i][j] -
(float)source_array3[i][j])
                                                                /(float)STEPS;
                step_array4[i][j] = ((float)target_array4[i][j] -
(float)source_array4[i][j])
                                                                /(float)STEPS;
                step_array5[i][j] = ((float)target_array5[i][j] -
(float)source_array5[i][j])
                                                                /(float)STEPS;
                */
        }
        toggle = 1;
        syssgi(SGI_SSYNC);
        syssgi(SGI_BDFLUSHCNT, 20);
        blip(ON);
        blip(OFF);
        for(s=0; s<STEPS; s++)
```

```
                {
                        for(i=0; i< TARGET_SIZE; i++)
                        for(j=0; j< TARGET_SIZE; j++)
                        {
                                current_array1[i][j] = current_array1[i][j] +
step_array1[i][j];

                                /*
                                current_array2[i][j] = current_array2[i][j] +
step_array2[i][j];
                                current_array3[i][j] = current_array3[i][j] +
step_array3[i][j];
                                current_array4[i][j] = current_array4[i][j] +
step_array4[i][j];
                                current_array5[i][j] = current_array5[i][j] +
step_array5[i][j];
                                */

                                show_array1[i][j] = (short)(current_array1[i][j] + 0.5);

                                /*
                                show_array2[i][j] = (short)(current_array2[i][j] + 0.5);
                                show_array3[i][j] = (short)(current_array3[i][j] + 0.5);
                                show_array4[i][j] = (short)(current_array4[i][j] + 0.5);
                                show_array5[i][j] = (short)(current_array5[i][j] + 0.5);
                                */

                                show_array1[i][j] = max(show_array1[i][j], 256);

                                /*
                                show_array2[i][j] = max(show_array2[i][j], 256);
                                show_array3[i][j] = max(show_array3[i][j], 256);
                                show_array4[i][j] = max(show_array4[i][j], 256);
                                show_array5[i][j] = max(show_array5[i][j], 256);
                                */

                                show_array1[i][j] = min(show_array1[i][j], 511);

                                /*
                                show_array2[i][j] = min(show_array2[i][j], 511);
                                show_array3[i][j] = min(show_array3[i][j], 511);
                                show_array4[i][j] = min(show_array4[i][j], 511);
                                show_array5[i][j] = min(show_array5[i][j], 511);
                                */
                        }
                        switch(TARGETS)
                        {
                                /*
                                case 5:
                                        rectwrite(x5,y5,x5-1+TARGET_SIZE, y5-1+TARGET_SIZE,
show_array5);
                                case 4:
                                        rectwrite(x4,y4,x4-1+TARGET_SIZE, y4-1+TARGET_SIZE,
show_array4);
                                case 3:
                                        rectwrite(x3,y3,x3-1+TARGET_SIZE, y3-1+TARGET_SIZE,
show_array3);
                                case 2:
                                        rectwrite(x2,y2,x2-1+TARGET_SIZE, y2-1+TARGET_SIZE,
show_array2);
                                */
                                case 1:
                                default:
                                        rectwrite(x1,y1,x1-1+TARGET_SIZE, y1-1+TARGET_SIZE,
show_array1);
                        }
                        if(qtest())
                        {
                                q_value = qread(&response);
                                if(response == DOWN)
                                {
                                        blip(ON);
```

```
                                        blip(OFF);
                                        switch(q_value)
                                        {
                                                case LEFTMOUSE:
                                                        stop_flag = 1;
                                                        break;
                                                case MIDDLEMOUSE:
                                                        stop_flag = 1;
                                                        break;
                                                case RIGHTMOUSE:
                                                        stop_flag = 1;
                                                        break;
                                                case SPACEKEY:
                                                        stop_flag = 1;
                                                        break;
                                                case ESCKEY:
                                                        fclose(outfile);
                                                        exit(-1);
                                                        break;
                                                default:
                                                        printf("Error: received unknown value
in queue.\n");
                                                        break;
                                        }
                                }
                                qreset();
                        }
                        if(stop_flag)
                        {
                                break;
                        }
                }
                if(gettimeofday(&tp,0))
                {
                        printf("End time call failed!\n\n");
                }
                end = tp.tv_sec;
                printf("Elapsed time for ramp = %d\n\n", end-start);
                switch(TARGETS)
                {
                        /*
                        case 5:
                                rectwrite(x5,y5,x5-1+TARGET_SIZE, y5-1+TARGET_SIZE,
target_array5);
                        case 4:
                                rectwrite(x4,y4,x4-1+TARGET_SIZE, y4-1+TARGET_SIZE,
target_array4);
                        case 3:
                                rectwrite(x3,y3,x3-1+TARGET_SIZE, y3-1+TARGET_SIZE,
target_array3);
                        case 2:
                                rectwrite(x2,y2,x2-1+TARGET_SIZE, y2-1+TARGET_SIZE,
target_array2);
                        */
                        case 1:
                        default:
                                rectwrite(x1,y1,x1-1+TARGET_SIZE, y1-1+TARGET_SIZE,
target_array1);
                }

                syssgi(SGI_SSYNC);
                fprintf(outfile, "%d\t%d\t%d\n", x1-128, y1, s);
                if(!stop_flag)
                {
                        do
                        {
                                while(!qtest());                                /* Wait to
stop trial     */
                                q_value = qread(&response);
                        }while(response != DOWN);
                        qreset();
                        switch(q_value)
```

```
                        {
                                case LEFTMOUSE:
                                        break;
                                case MIDDLEMOUSE:
                                        break;
                                case RIGHTMOUSE:
                                        break;
                                case SPACEKEY:          /* Allow operator to end trial
        */
                                        fprintf(outfile, "# Operator Terminated Trial\n\n");
                                        break;
                                case ESCKEY:
                                        fclose(outfile);
                                        exit(-1);
                                        break;
                                default:
                                        printf("Error: received unknown value in queue.\n");
                                        break;
                        }
                        qreset();
                }
                else
                {
                        stop_flag = 0;
                        ticks_left = 300;
                        while(ticks_left)
                        {
                                ticks_left = sginap(ticks_left);
                        }

                        while(qtest())
                        {
                                qreset();
                        }
                }
                switch(TARGETS)
                {
                        /*
                        case 5:
                                rectwrite(x5,y5,x5-1+TARGET_SIZE, y5-1+TARGET_SIZE,
source_array5);
                        case 4:
                                rectwrite(x4,y4,x4-1+TARGET_SIZE, y4-1+TARGET_SIZE,
source_array4);
                        case 3:
                                rectwrite(x3,y3,x3-1+TARGET_SIZE, y3-1+TARGET_SIZE,
source_array3);
                        case 2:
                                rectwrite(x2,y2,x2-1+TARGET_SIZE, y2-1+TARGET_SIZE,
source_array2);
                        */
                        case 1:
                        default:
                                rectwrite(x1,y1,x1-1+TARGET_SIZE, y1-1+TARGET_SIZE,
source_array1);
                }

                swapbuffers();          /* Display Fixation Buffer          */
                trials++;
        }
    }
}
/*-------------------------------------------------------------------------*/
                        /***************
                        *  graph_init  *
                        ***************/

graph_init()
{
    char    color_string[10];
    int     i,
                color_value;
    long    type;
```

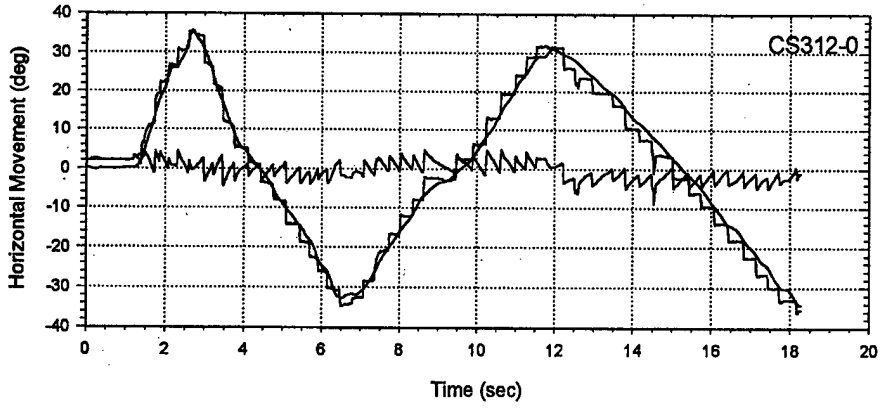
```
        FILE    *fp;
        prefposition(0,XMAXSCREEN,0,YMAXSCREEN);
        winopen("image");
        sleep(2);
        cmode();
        doublebuffer();
        gconfig();
        cursoff();          /* Turn the cursor off. */

        backbuffer(TRUE);
        frontbuffer(FALSE);
        qdevice(LEFTMOUSE);
        qdevice(MIDDLEMOUSE);
        qdevice(RIGHTMOUSE);
        qdevice(ESCKEY);
        qdevice(SPACEKEY);
        save_color_map();
        type = getmonitor();
        /* If a projector calibration file is not found in my home directory,
               load a linear colormap. */
        if(access("/usr/people/craig/barco.cal", 0) != NULL)
        {
               for(i=0; i<256; i++)
               {
                      mapcolor(256+i, i, i, i);
               }
        }
        else
        {
               fp = fopen("/usr/people/craig/barco.cal", "r");
               for (i=0; i<256; i++)
               {
                      fgets(color_string, 10, fp);
                      color_value = atoi(color_string);
                      mapcolor(256+i, color_value, color_value, color_value);
               }
        }
        /* Initialize the color used to signal the timing sampler. */
        mapcolor(SIGNAL_COLOR, 0, 0, 0);
}
/*-------------------------------------------------------------------------*/
                                /**************
                                *  init_blip  *
                                **************/
init_blip()
{
        int i, j;
        /* Initialize the color used to signal the timing sampler. */
        mapcolor(SIGNAL_COLOR, 0, 0, 0);
        for(i=0; i<BLIP_SIZE; i++)
        for(j=0; j<BLIP_SIZE; j++)
        {
               blip_array[i][j] = SIGNAL_COLOR;
        }
        rectwrite(32, 32, 31+BLIP_SIZE, 31+BLIP_SIZE, blip_array);
}
/*-------------------------------------------------------------------------*/
                                /***********************
                                *  init_current_array  *
                                ***********************/
init_current_array()
{
        int     i,j;
        for(i=0; i<TARGET_SIZE; i++)
        for(j=0; j<TARGET_SIZE; j++)
        {
               current_array1[i][j] = (float)source_array1[i][j];

               /*
               current_array2[i][j] = (float)source_array2[i][j];
               current_array3[i][j] = (float)source_array3[i][j];
               current_array4[i][j] = (float)source_array4[i][j];
```

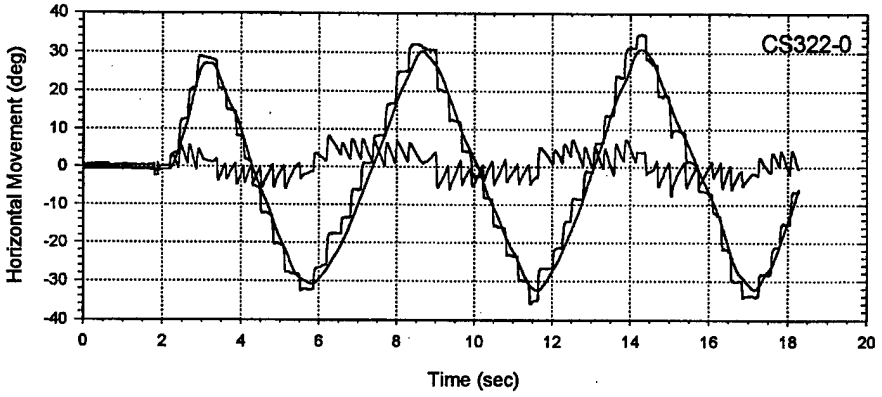
```
                current_array5[i][j] = (float)source_array5[i][j];
                */
        }
}
/*-------------------------------------------------------------------------*/
                                /**********************
                                *  init_target_array  *
                                **********************/
init_target_array()
{
        int     i,j;
        float   target_temp,
                        target_tempb,
                        source_temp,
                        shift_orig,
                        radius,
                        gaussian,
                        x,y;
        shift_orig = (TARGET_SIZE - 1.0) / 2.0;
        for(i=0; i<TARGET_SIZE; i++)
        for(j=0; j<TARGET_SIZE; j++)
        {
                x = (float)(j - shift_orig);
                y = (float)(i - shift_orig);
                radius = sqrt(x*x + y*y);
                gaussian = exp(-((4.0*radius/TARGET_SIZE)*(4.0*radius/TARGET_SIZE)));
                /* Add 1 to the sine value to shift its range to 0-2.  Divide color by 2 to adjust
for
                        sine shift.  Add 256 to get to correct starting location in colormap.  */
/*              target_temp = (255.0/2.0) * (cos(j*2.0*M_PI/PPC) + 1.00) + 256;  */
/*              target_temp = (255.0/4.0) * (sin(j*2.0*M_PI/PPC) + 1.00) + 256;  */
                target_temp = exp(-M_PI*((x/D)*(x/D) + (y/D)*(y/D)));
                target_temp = 1.0;
/*              target_temp = (255.0/2.0)*(target_temp * (cos(2.0*M_PI*Fx*x/N + 2.0*M_PI*Fy*y/N +
PHx + PHy) + 1.0)) + 256.0; */
                Fx = F * cos(Theta);
                Fy = F * sin(Theta);
                target_temp = (255.0/2.0)*(target_temp * (cos(2.0*M_PI*Fx*x/N + 2.0*M_PI*Fy*y/N +
PHx + PHy) + 1.0)) + 256.0;
                target_tempb = (255.0/2.0)*(1.0 * (cos(2.0*M_PI*Fx*x/N + 3.0*M_PI*Fy*y/N + PHx +
PHy) + 1.0)) + 256.0;
/*              target_temp = target_temp + (255.0/4.0) * (sin(i*2.0*M_PI/PPC) + 1.00) + 256;*/
                target_temp = gaussian * target_temp;
                target_tempb = gaussian * target_tempb;
                source_temp = (1.0 - gaussian) * source_array1[i][j];
                target_array1[i][j] = (short)(target_temp + source_temp + 0.5);

                /*
                source_temp = (1.0 - gaussian) * source_array2[i][j];
                target_array2[i][j] = (short)(target_temp + source_temp + 0.5);
                source_temp = (1.0 - gaussian) * source_array3[i][j];
                target_array3[i][j] = (short)(target_temp + source_temp + 0.5);
                source_temp = (1.0 - gaussian) * source_array4[i][j];
                target_array4[i][j] = (short)(target_temp + source_temp + 0.5);
                source_temp = (1.0 - gaussian) * source_array5[i][j];
                target_array5[i][j] = (short)(target_tempb + source_temp + 0.5);
                */
        }
}
/*-------------------------------------------------------------------------*/
                                /********************
                                *  preview_target  *
                                ********************/
preview_target()
{
        int     x,
                        y;
        long    ticks_left;

        x = X_CENTER - (TARGET_SIZE/2);
        y = Y_CENTER - (TARGET_SIZE/2);
```

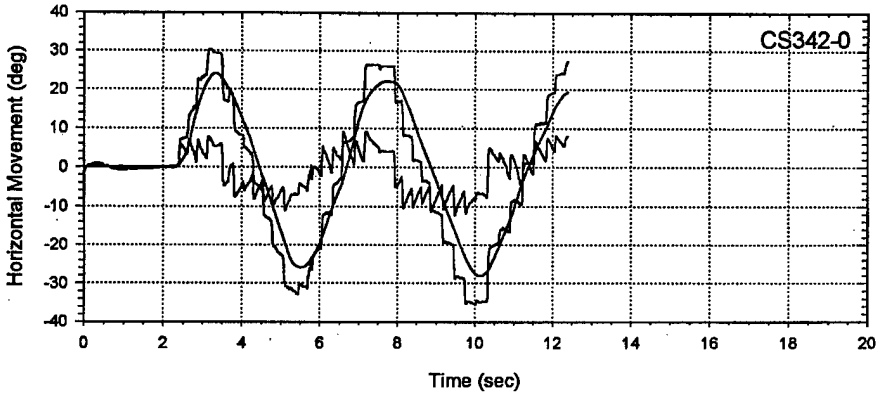
```
        rectread(x, y, (x-1+TARGET_SIZE),(y-1+TARGET_SIZE),source_array1);
        init_target_array();
        rectwrite(x,y,x-1+TARGET_SIZE, y-1+TARGET_SIZE, target_array1);
        ticks_left = sginap(PREVIEW_TIME * CLK_TCK);

        while(qtest())
        {
                qreset();
        }
        rectwrite(x,y,x-1+TARGET_SIZE, y-1+TARGET_SIZE, source_array1);
}
/*---------------------------------------------------------------------------*/
                                /************************
                                *  randomize_locations  *
                                ************************/
randomize_locations(int locations)
{
        int     i,
                        j,
                        random_value = 0;
        long    seed_value;
        int     temp_x,
                        temp_y;

        seed_value = (long)time(0);
        srand(seed_value);
        for(j=0; j<10; j++)
        {
                for(i=0; i<locations; i++)
                {
                        random_value =  rand() % (locations);

                        temp_x = target_locs[0][i];
                        temp_y = target_locs[1][i];

                        target_locs[0][i] = target_locs[0][random_value];
                        target_locs[1][i] = target_locs[1][random_value];

                        target_locs[0][random_value] = temp_x;
                        target_locs[1][random_value] = temp_y;
                }
        }
}                               /*** randomize_locations ***/
/*---------------------------------------------------------------------------*/
                                /**********************
                                *  restore_color_map  *
                                **********************/
restore_color_map()
{
        int     i;
        for(i=0; i<256; i++)
        {
                mapcolor(i+256, red[i], green[i], blue[i]);
        }
}
/*---------------------------------------------------------------------------*/
                                /*******************
                                *  save_color_map  *
                                *******************/
save_color_map()
{
        int     i;
        red = (short *)malloc(sizeof(short) * 256 * 3);
        green = red + 256;
        blue = green + 256;
        for(i=0; i<256; i++)
        {
                getmcolor(i+256, red+i, green+i, blue+i);
        }
}
          /*---------------------------------------------------------------------------*/
```

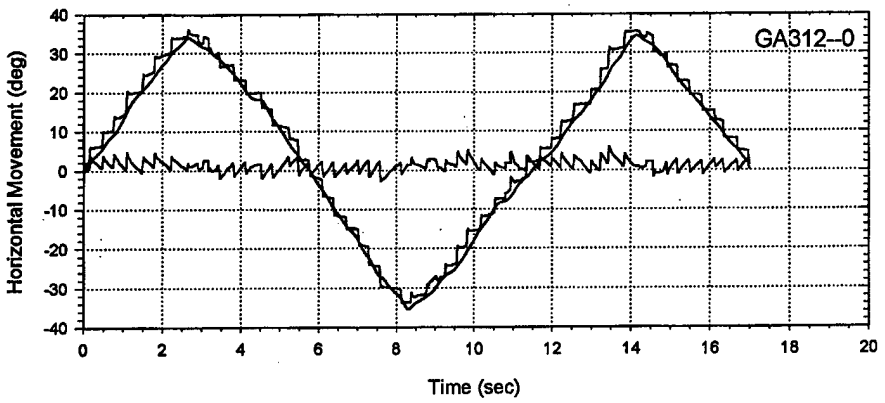
6.2 Appendix B

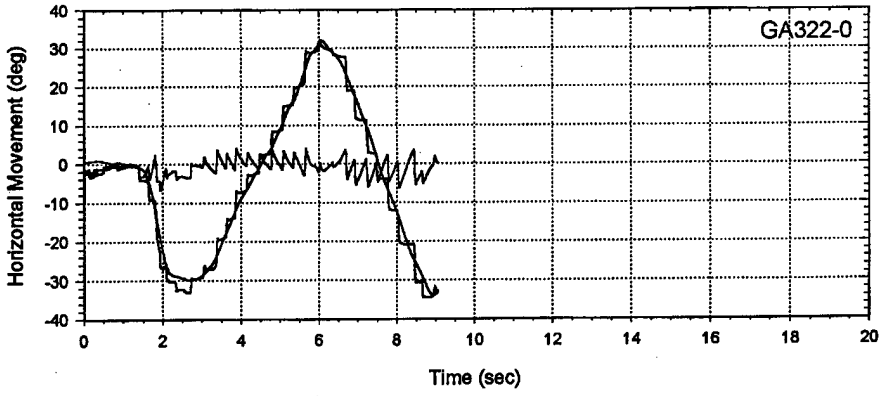
Head, eye, and gaze records for the observers tested under the IFOV condition using the 2° test stimulus (coded as "3" in the first numeric position in each graph label), the 'city' background image (coded as "2" in the third numeric position), and each of the three IFOVs of 10°, 20°, and 40° (coded as 1, 2, and 4, respectively, in the second numeric position).

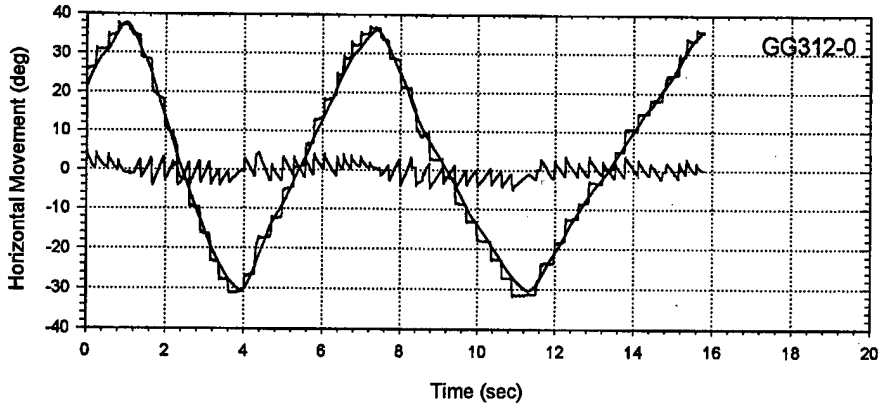

CS312-0
Horizontal Movement (deg)
Time (sec)

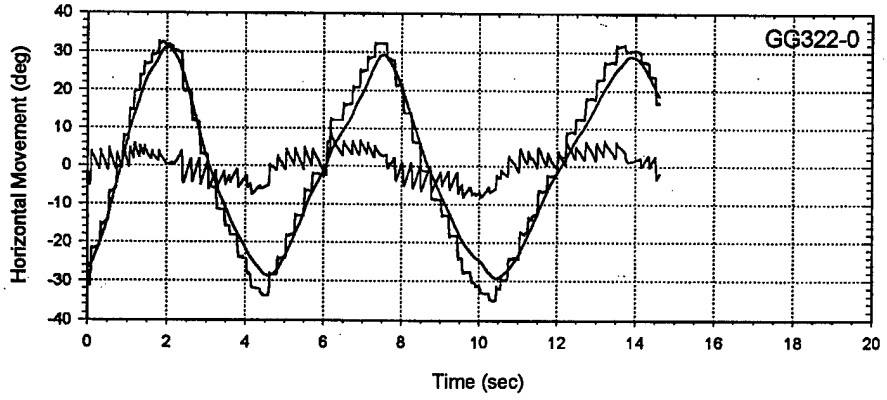

CS322-0
Horizontal Movement (deg)
Time (sec)

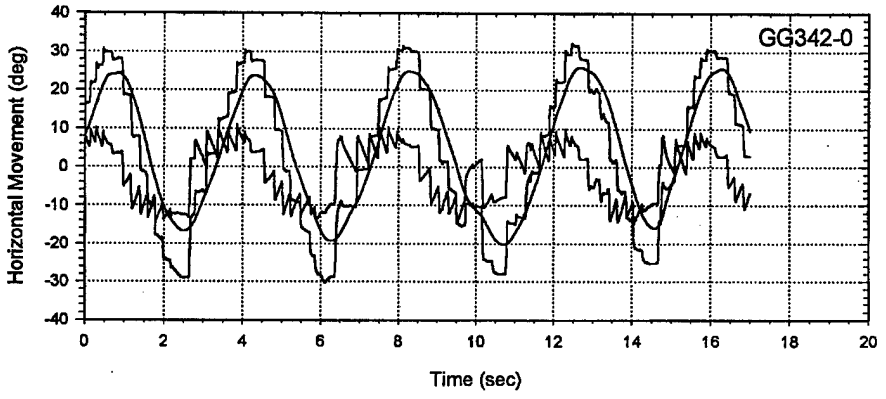

CS342-0
Horizontal Movement (deg)
Time (sec)

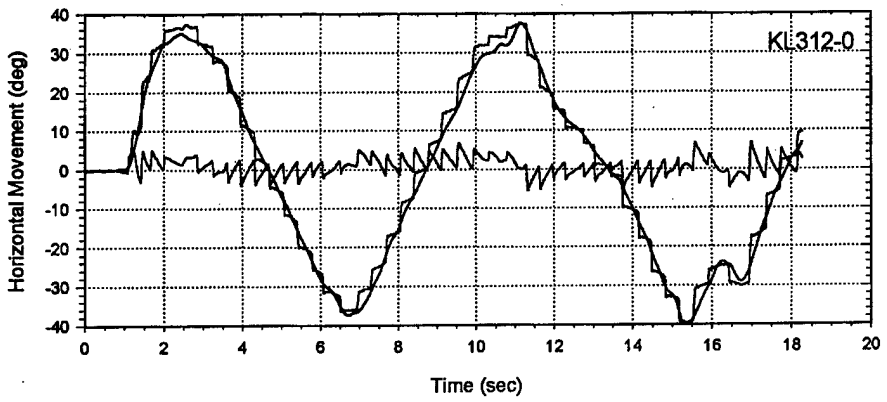

GA312-0
Horizontal Movement (deg)
40
30
20
10
0
-10
-20
-30
-40
0
2
4
6
8
10
12
14
16
18
20
Time (sec)

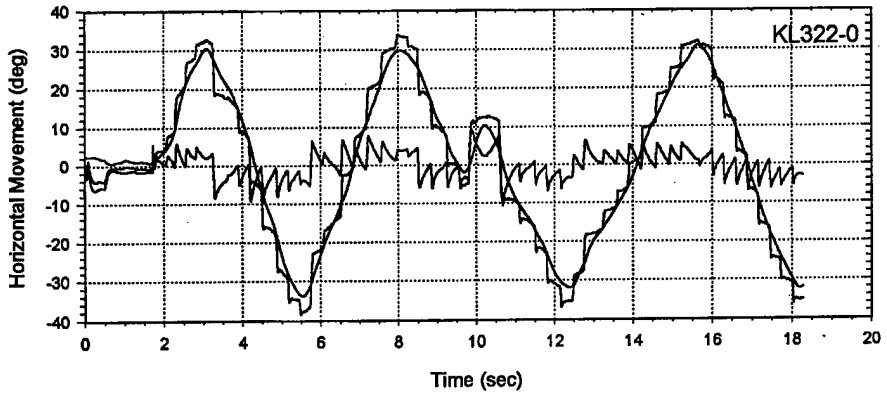

GA322-0
Horizontal Movement (deg)
40
30
20
10
0
-10
-20
-30
-40
0
2
4
6
8
10
12
14
16
18
20
Time (sec)

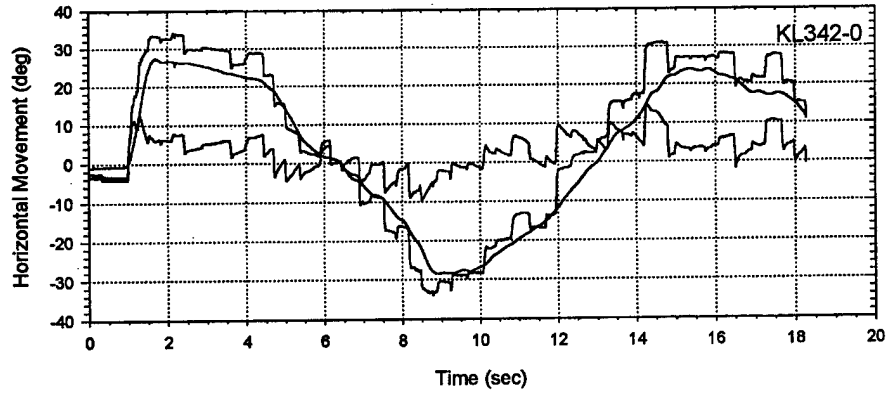

GG312-0
Horizontal Movement (deg)
Time (sec)

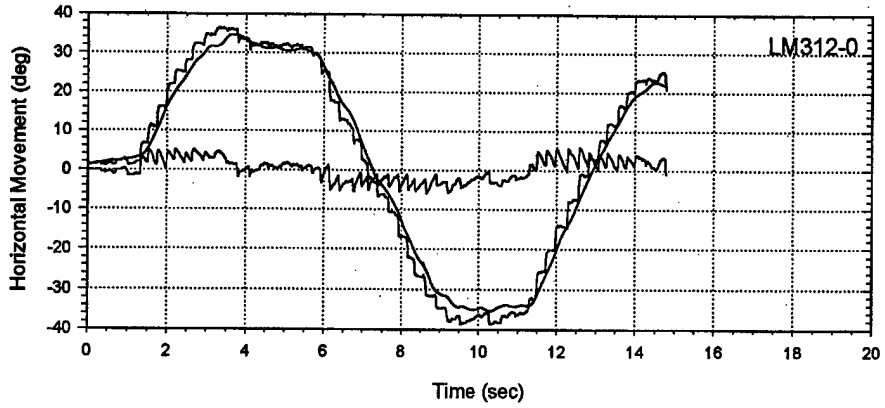

GG322-0
Horizontal Movement (deg)
Time (sec)

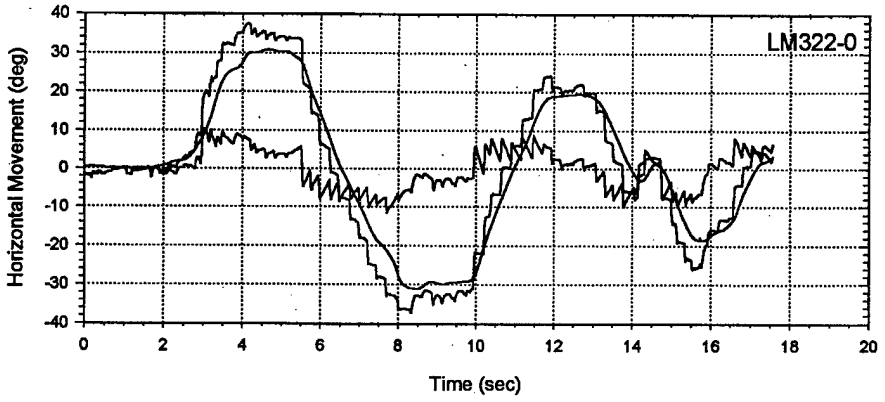

GG342-0
Horizontal Movement (deg)
Time (sec)

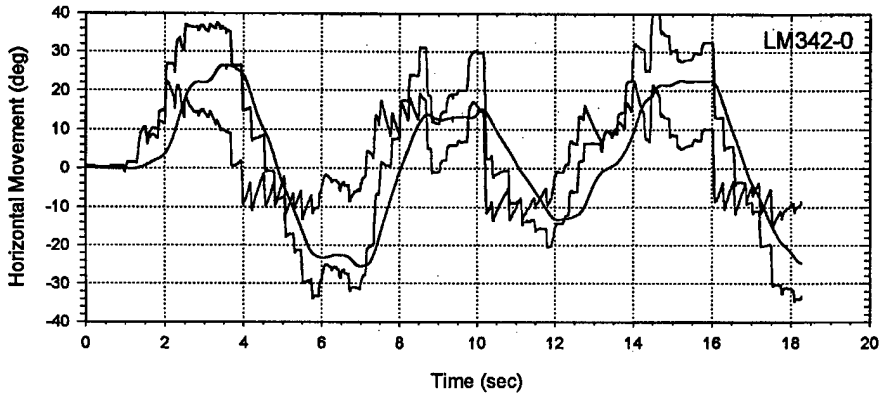

KL312-0
Horizontal Movement (deg)
Time (sec)

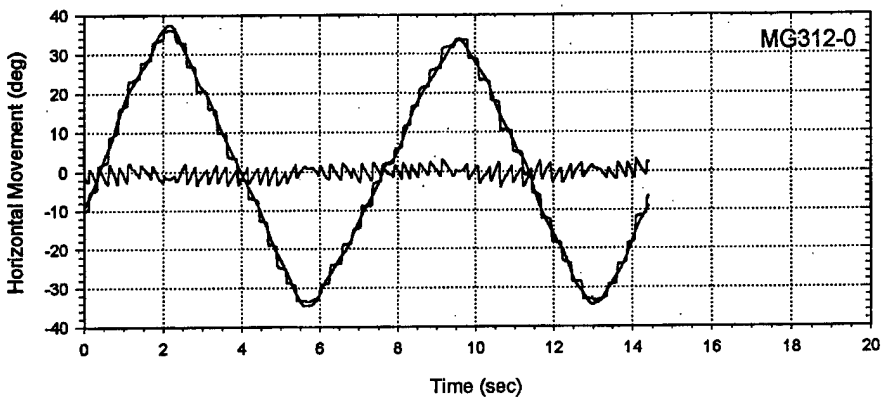

KL322-0
Horizontal Movement (deg)
Time (sec)

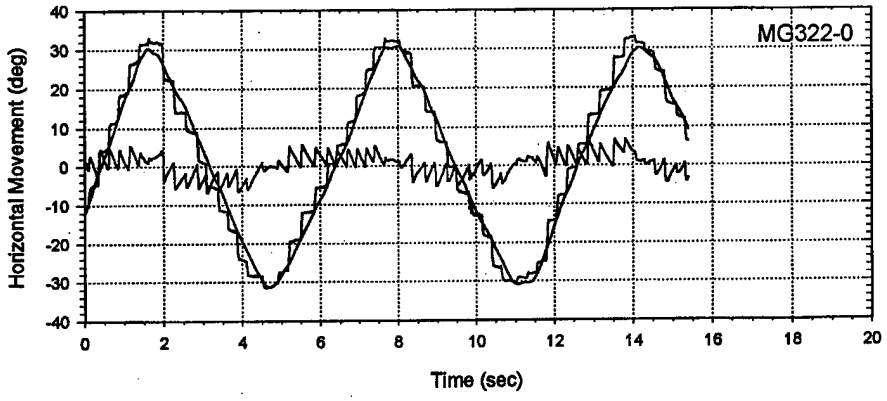

KL342-0
Horizontal Movement (deg)
Time (sec)

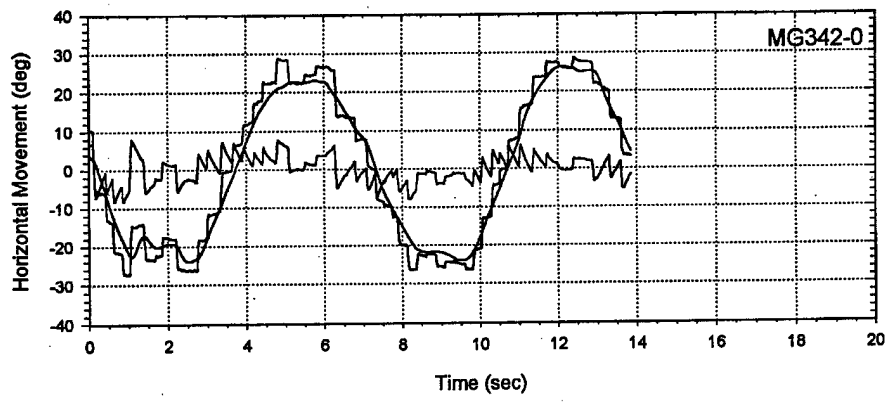

LM312-0
Horizontal Movement (deg)
Time (sec)


LM322-0
Horizontal Movement (deg)
Time (sec)


LM342-0
Horizontal Movement (deg)
Time (sec)


MG312-0
Horizontal Movement (deg)
Time (sec)


MG322-0
Horizontal Movement (deg)
Time (sec)

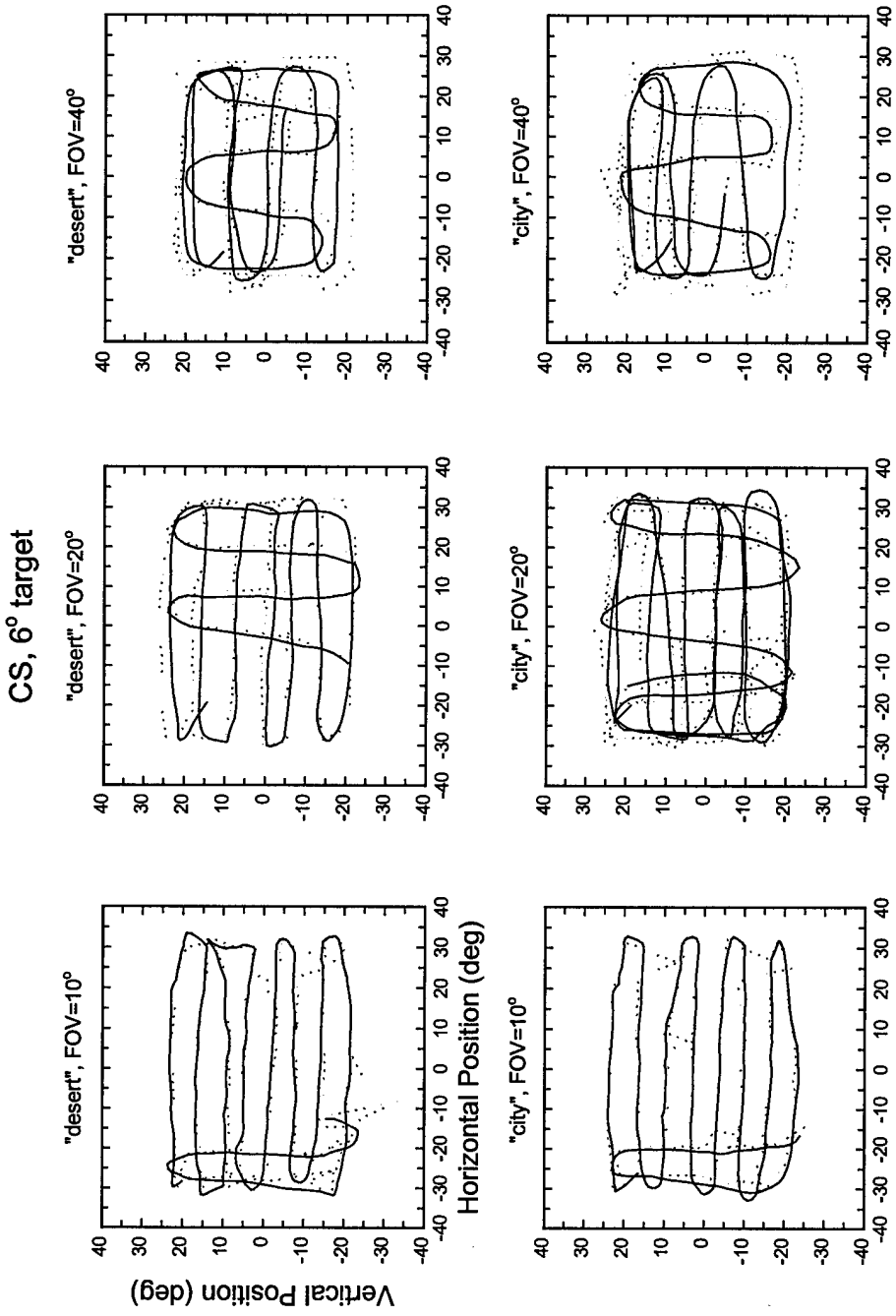

MG342-0
Horizontal Movement (deg)
Time (sec)

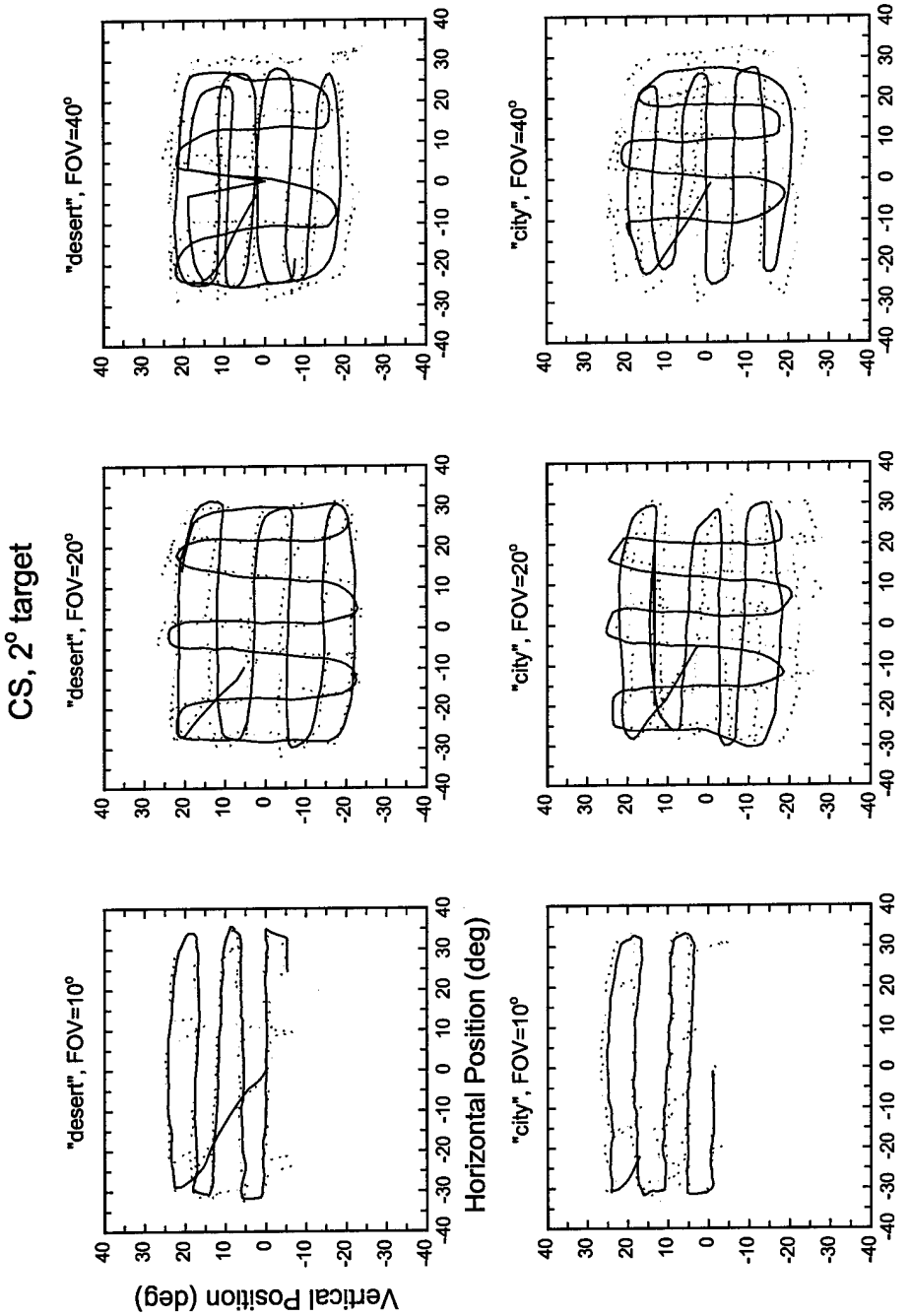
This page intentionally left blank.

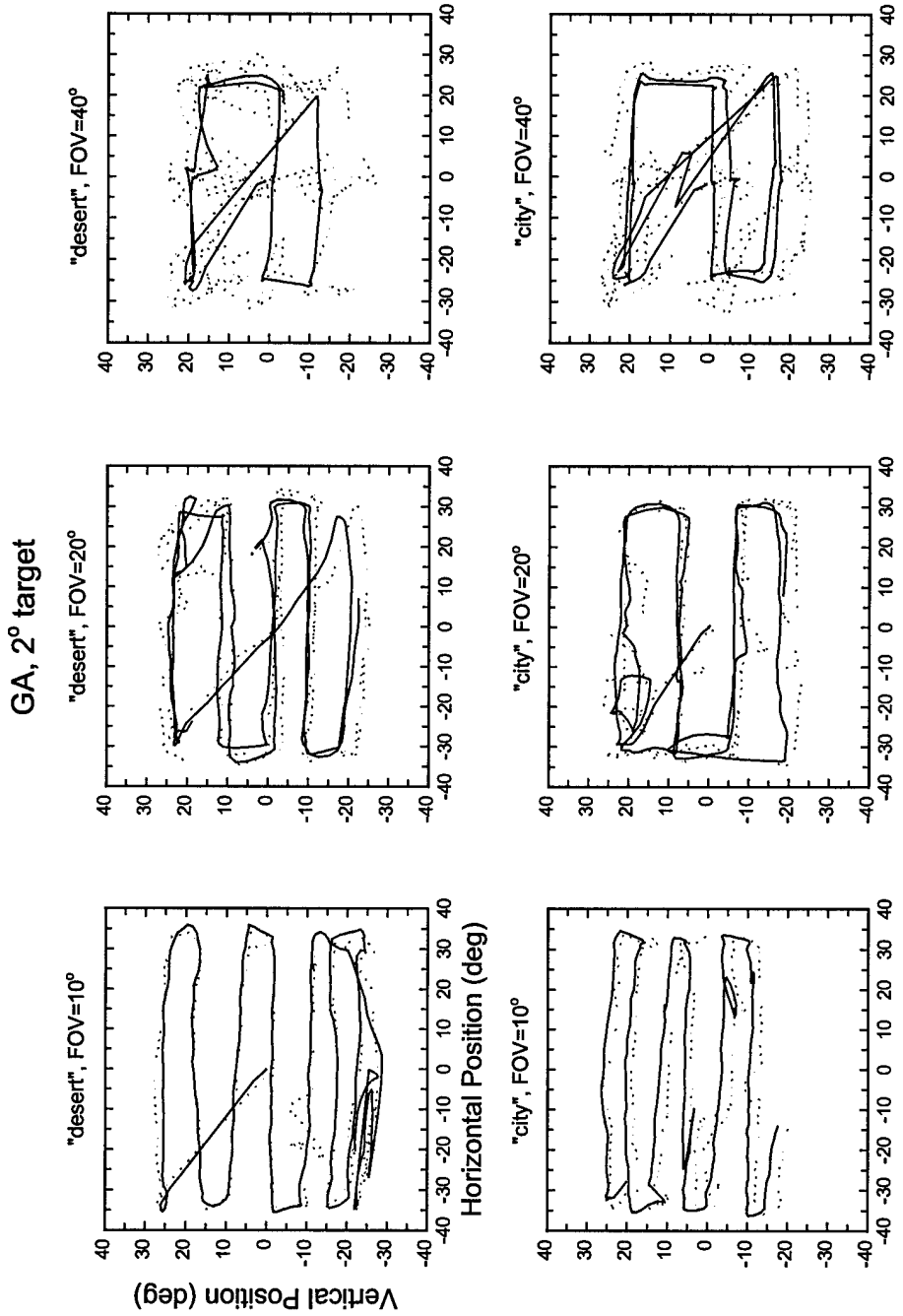
6.3 Appendix C

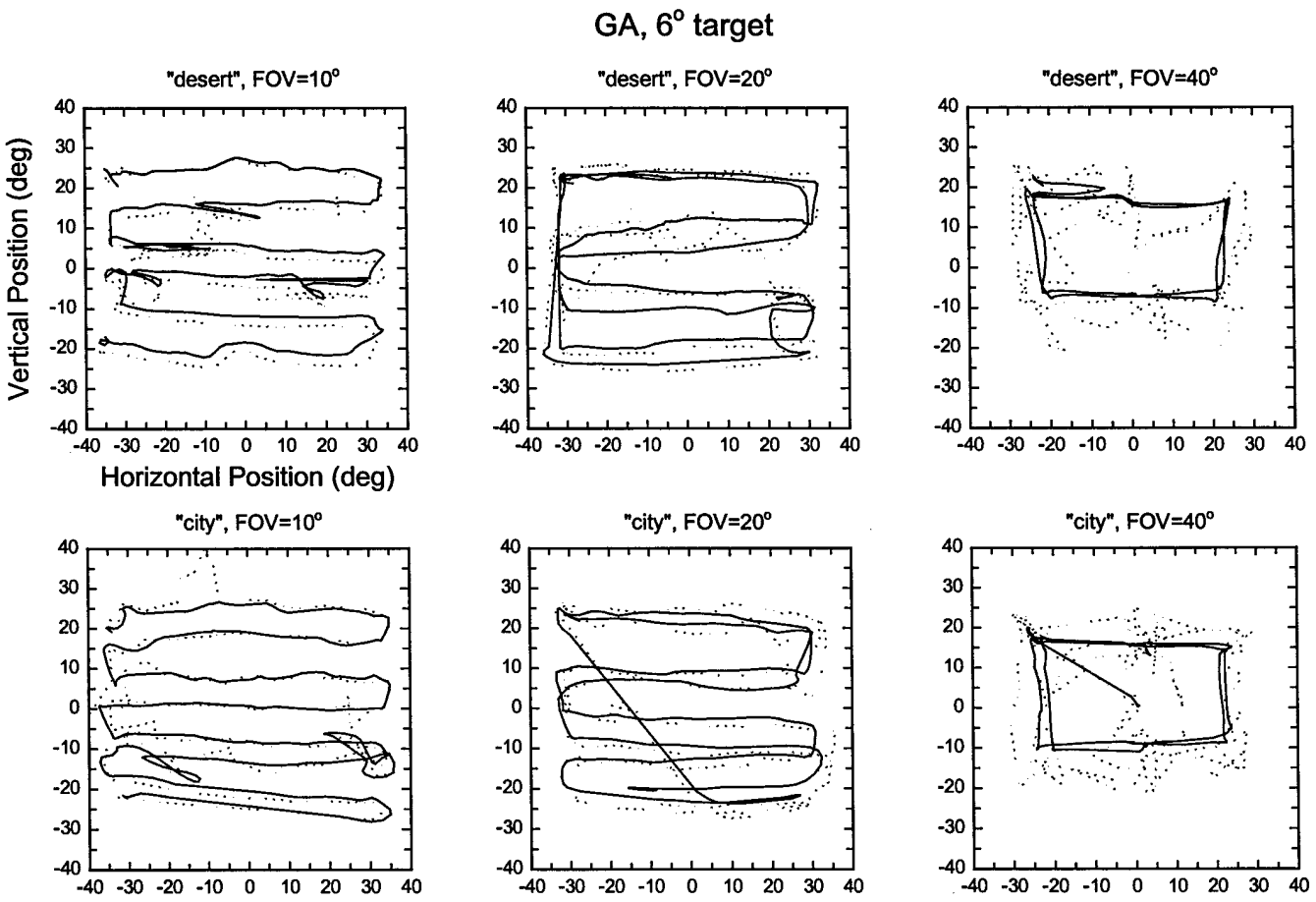
Head and gaze scanpaths obtained for one randomly selected trial from each of the seven observers tested in the IFOV condition. Scanpaths were obtained under each of the 12 (2 target sizes × 3 IFOVs × 2 background images) testing conditions.

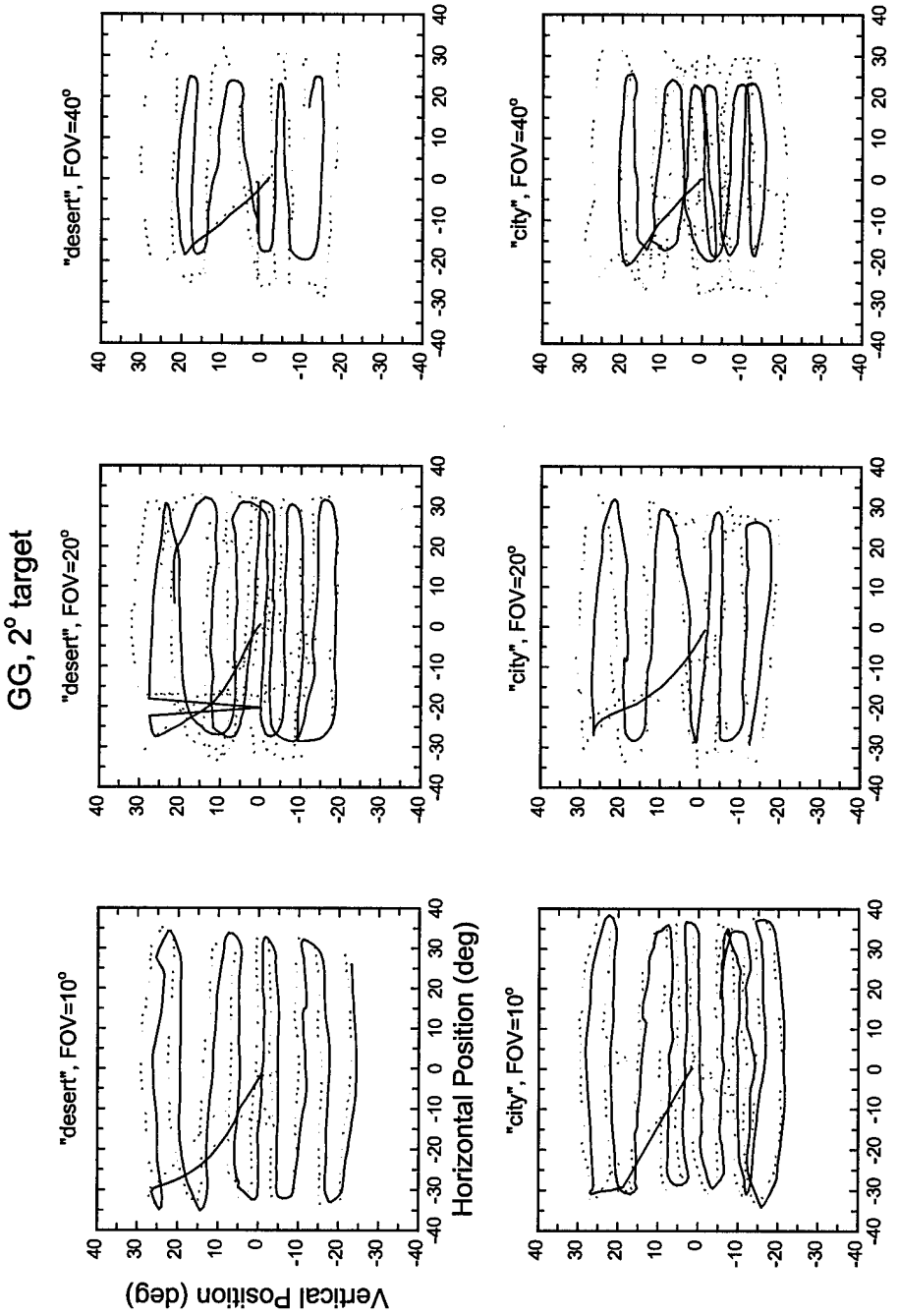
Head and gaze scanpaths obtained for one randomly selected trial from each of the five observers tested in the Real-NVG condition. Scanpaths were obtained under each of the four (2 target sizes × 2 background images) testing conditions.

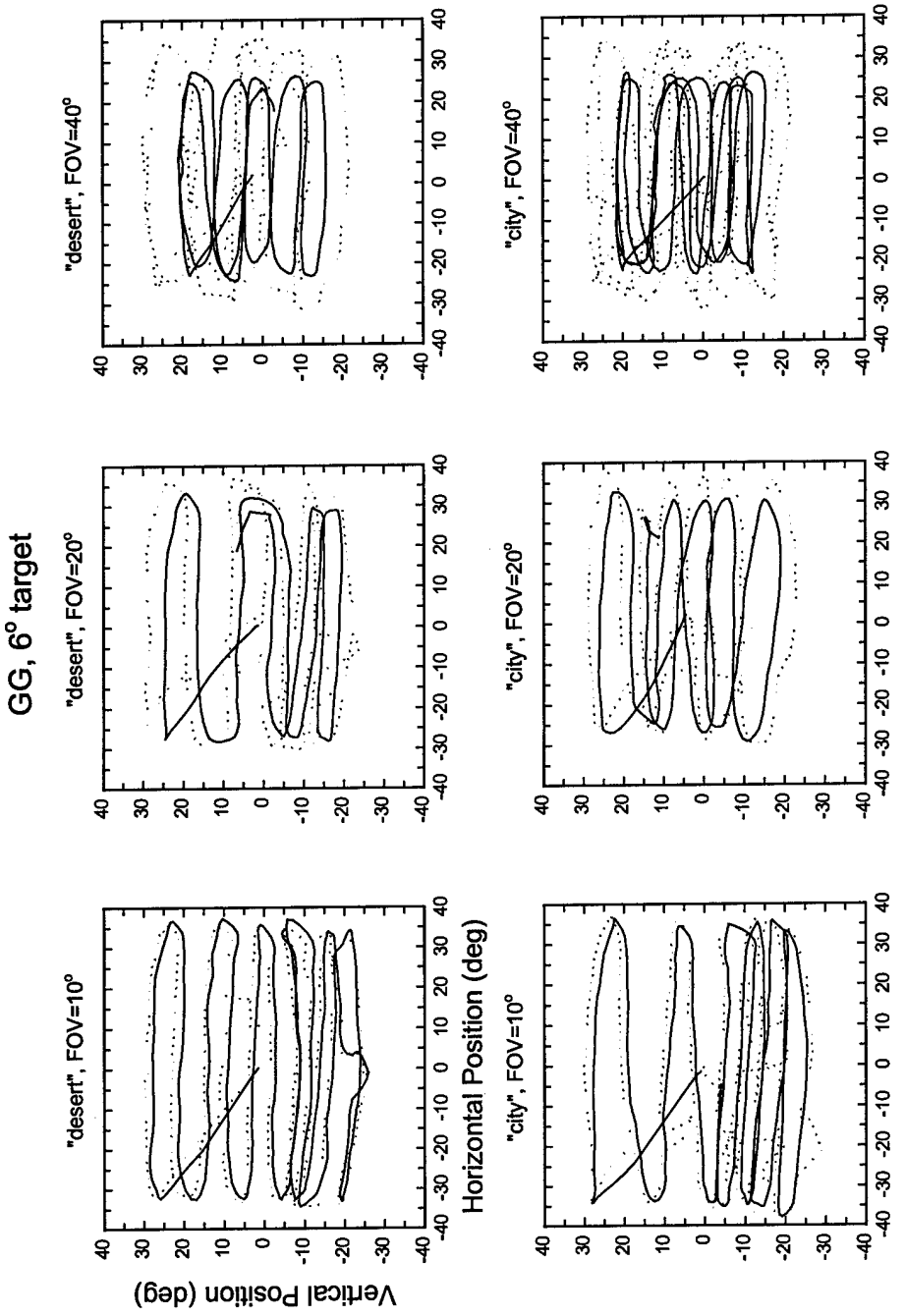

CS, 6° target
"desert", FOV=40°
"desert", FOV=20°
"desert", FOV=10°
"city", FOV=40°
"city", FOV=20°
"city", FOV=10°
Vertical Position (deg)
Horizontal Position (deg)

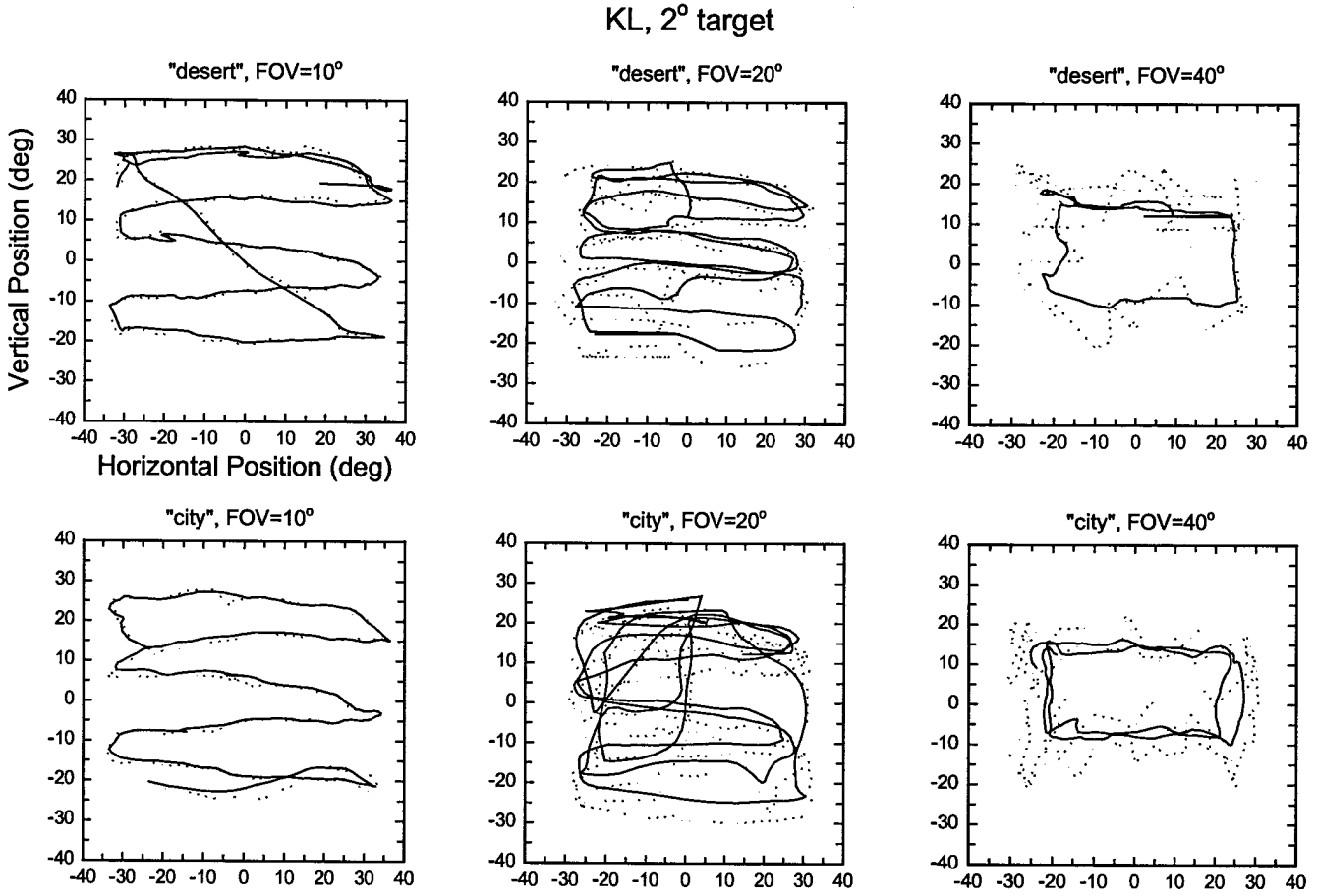

CS, 2° target
"desert", FOV=40°
"desert", FOV=20°
"desert", FOV=10°
"city", FOV=40°
"city", FOV=20°
"city", FOV=10°
Vertical Position (deg)
Horizontal Position (deg)

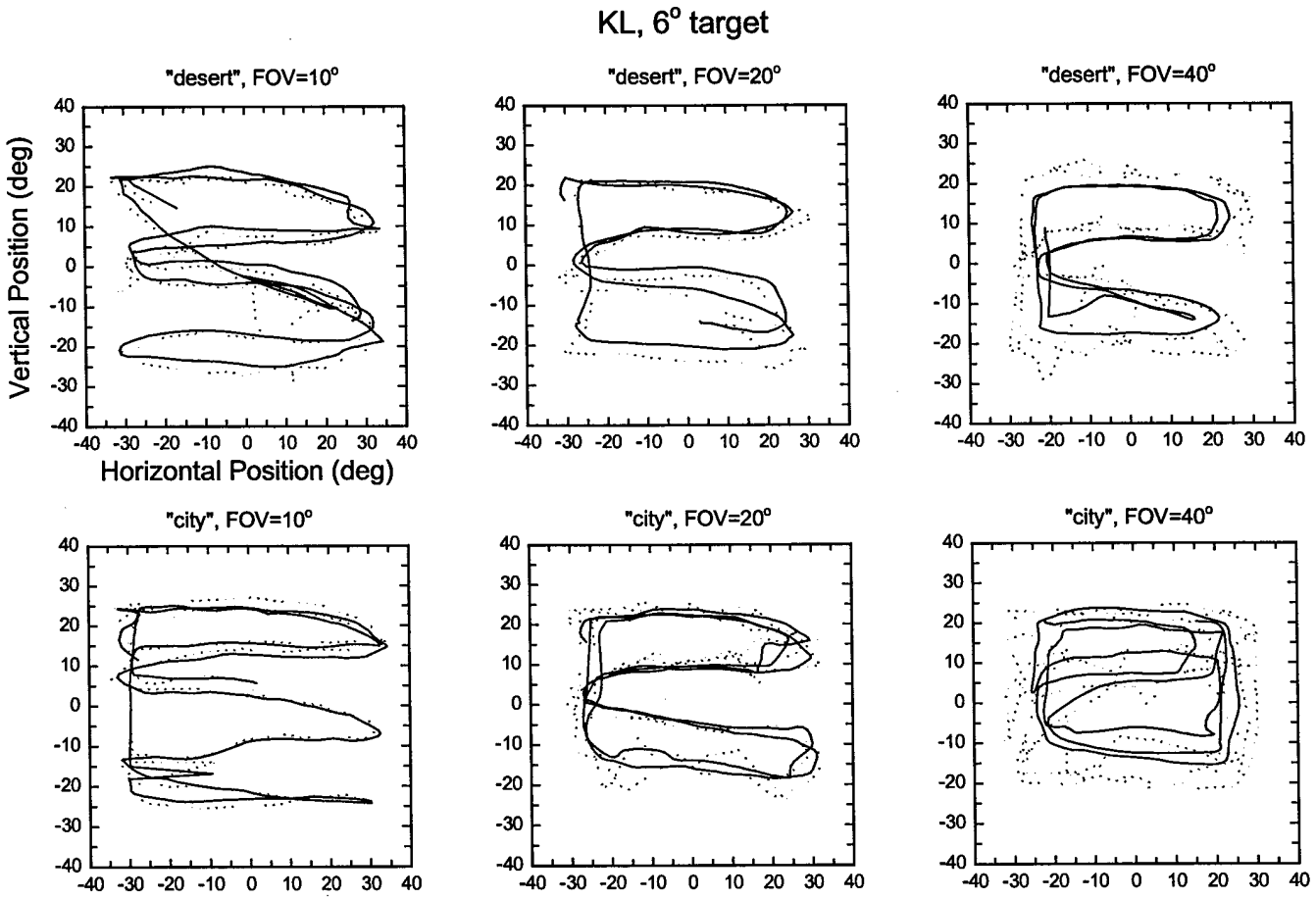

GA, 2° target
"desert", FOV=10°
"desert", FOV=20°
"desert", FOV=40°
"city", FOV=10°
"city", FOV=20°
"city", FOV=40°
Vertical Position (deg)
Horizontal Position (deg)
40
30
20
10
0
-10
-20
-30
-40

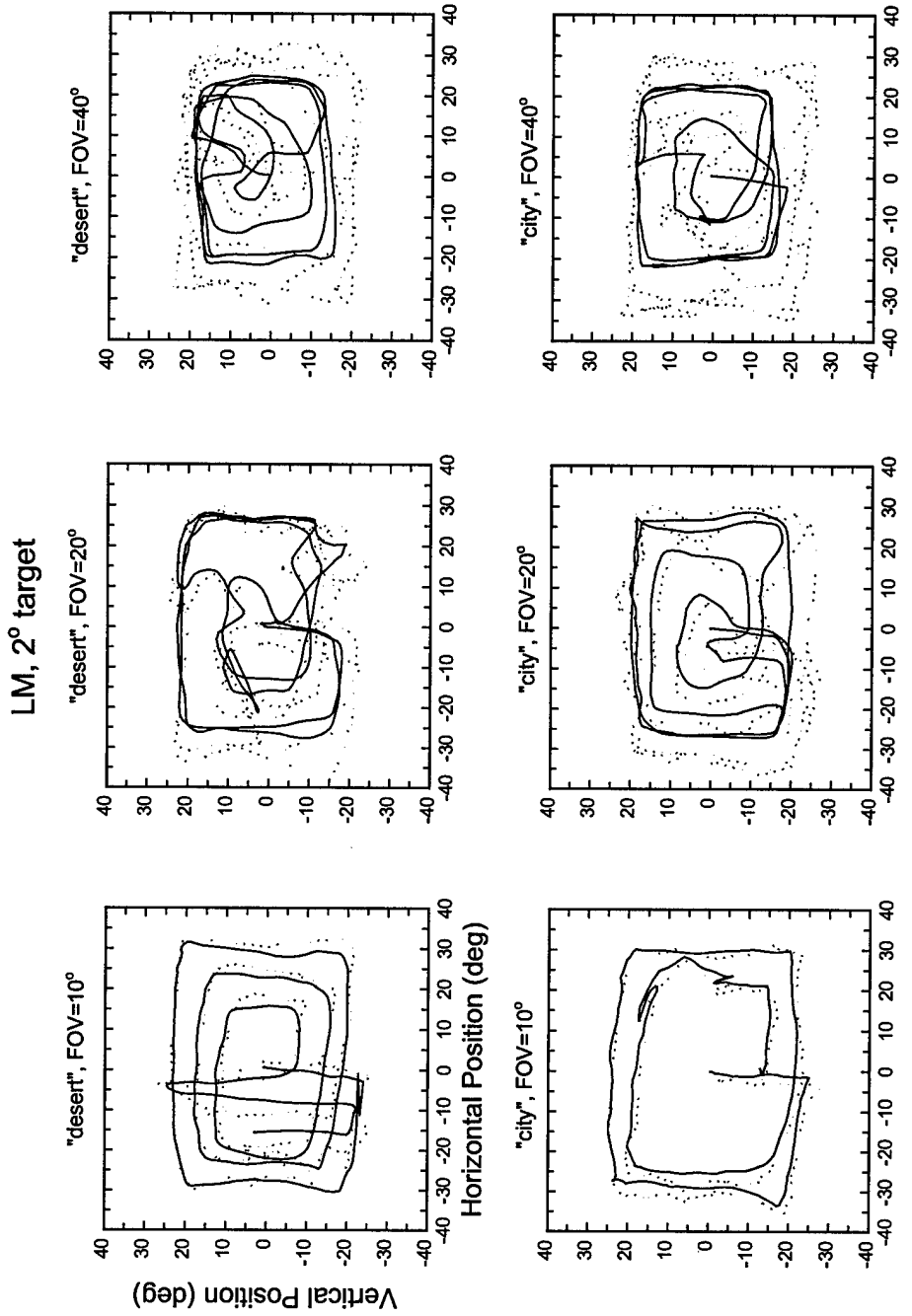

GA, 6° target
"desert", FOV=10°
"desert", FOV=20°
"desert", FOV=40°
"city", FOV=10°
"city", FOV=20°
"city", FOV=40°
Vertical Position (deg)
Horizontal Position (deg)
40
30
20
10
0
-10
-20
-30
-40
-40 -30 -20 -10 0 10 20 30 40

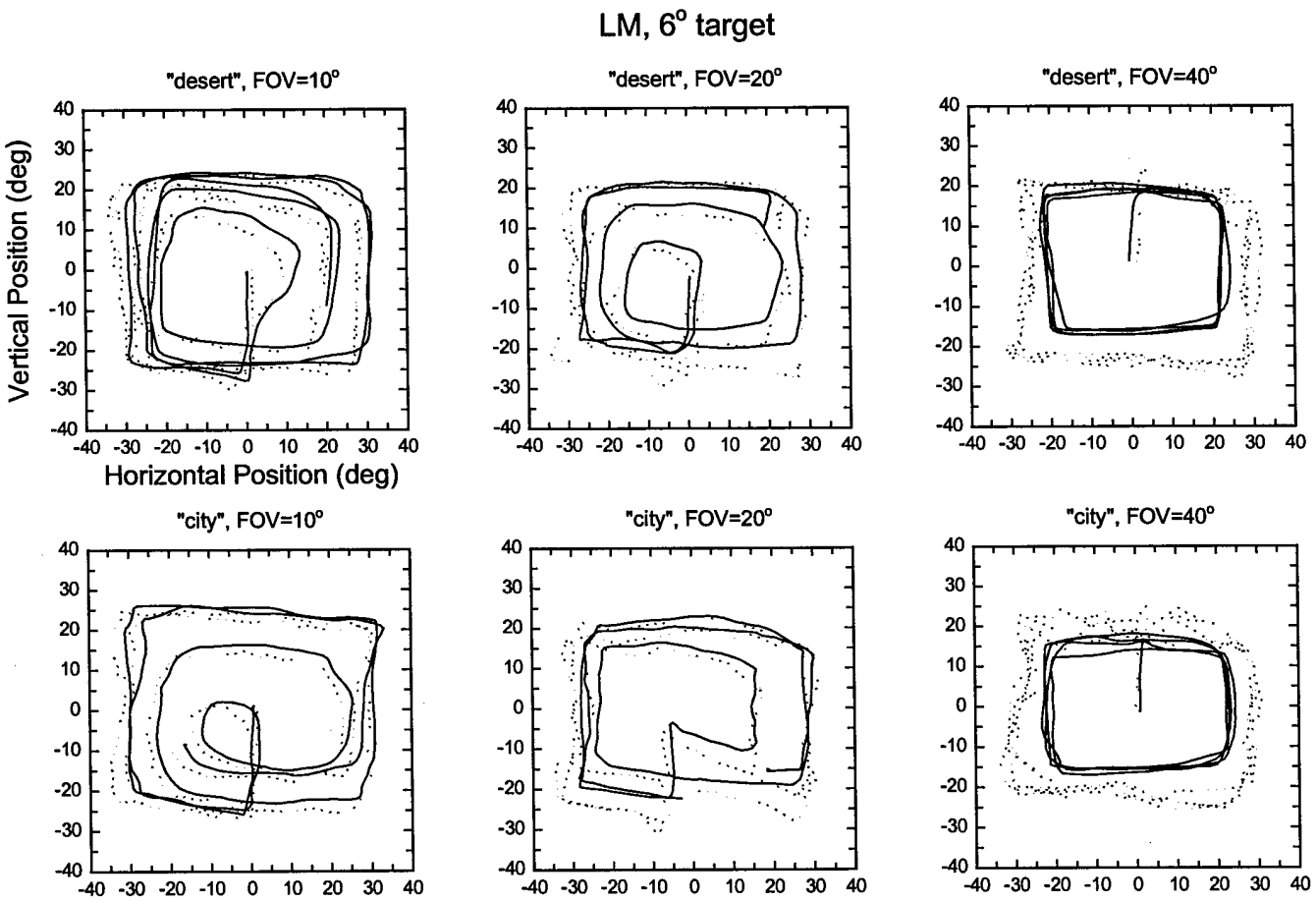

GG, 2° target
"desert", FOV=40°
"desert", FOV=20°
"desert", FOV=10°
"city", FOV=40°
"city", FOV=20°
"city", FOV=10°
Vertical Position (deg)
Horizontal Position (deg)

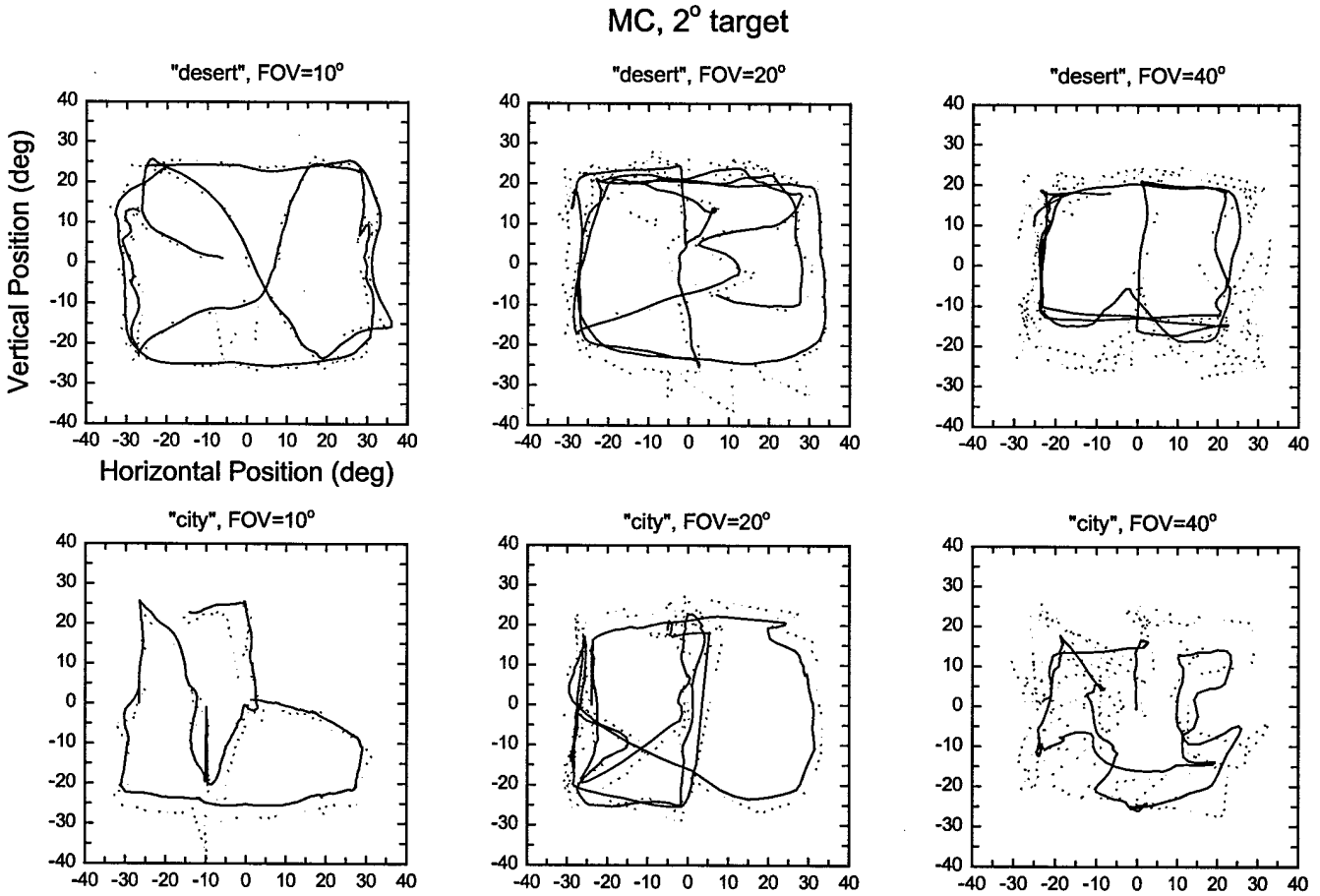

GG, 6° target
"desert", FOV=10°
"desert", FOV=20°
"desert", FOV=40°
"city", FOV=10°
"city", FOV=20°
"city", FOV=40°
Vertical Position (deg)
Horizontal Position (deg)

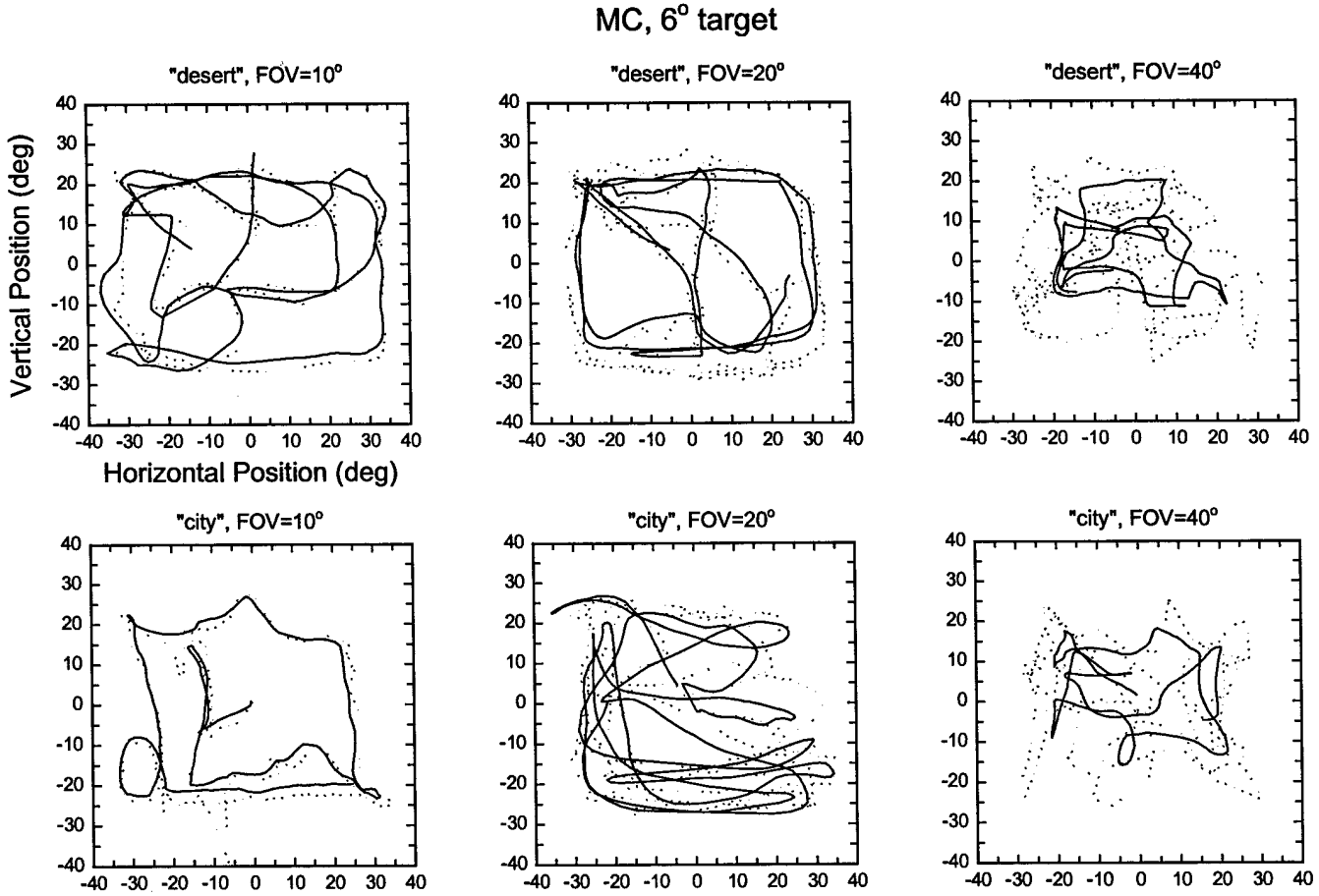

KL, 2° target
"desert", FOV=10°
"desert", FOV=20°
"desert", FOV=40°
"city", FOV=10°
"city", FOV=20°
"city", FOV=40°
Vertical Position (deg)
Horizontal Position (deg)
40
30
20
10
0
-10
-20
-30
-40
-40 -30 -20 -10 0 10 20 30 40

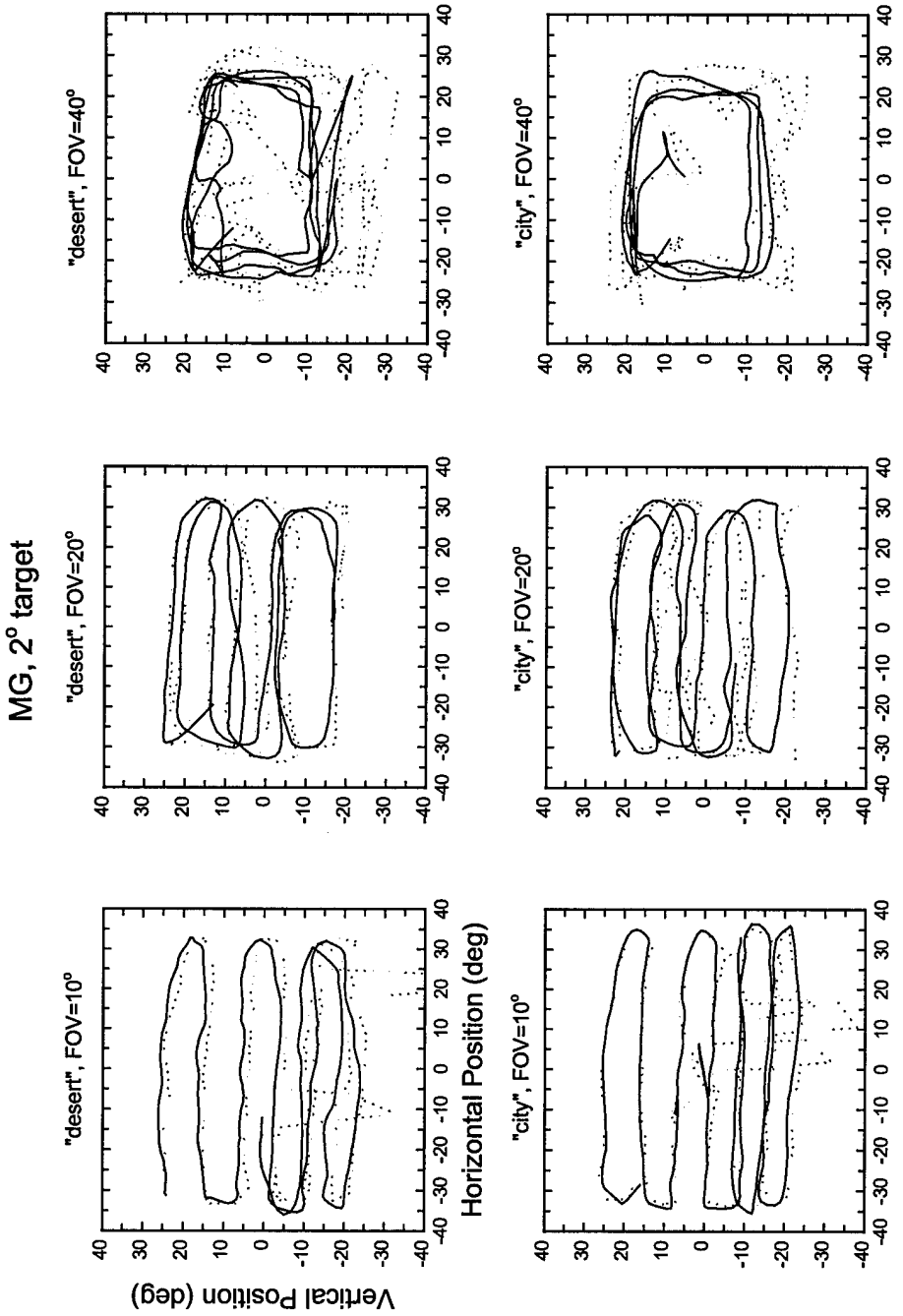

KL, 6° target
"desert", FOV=10°
"desert", FOV=20°
"desert", FOV=40°
"city", FOV=10°
"city", FOV=20°
"city", FOV=40°
Vertical Position (deg)
Horizontal Position (deg)
40
30
20
10
0
-10
-20
-30
-40
-40 -30 -20 -10 0 10 20 30 40

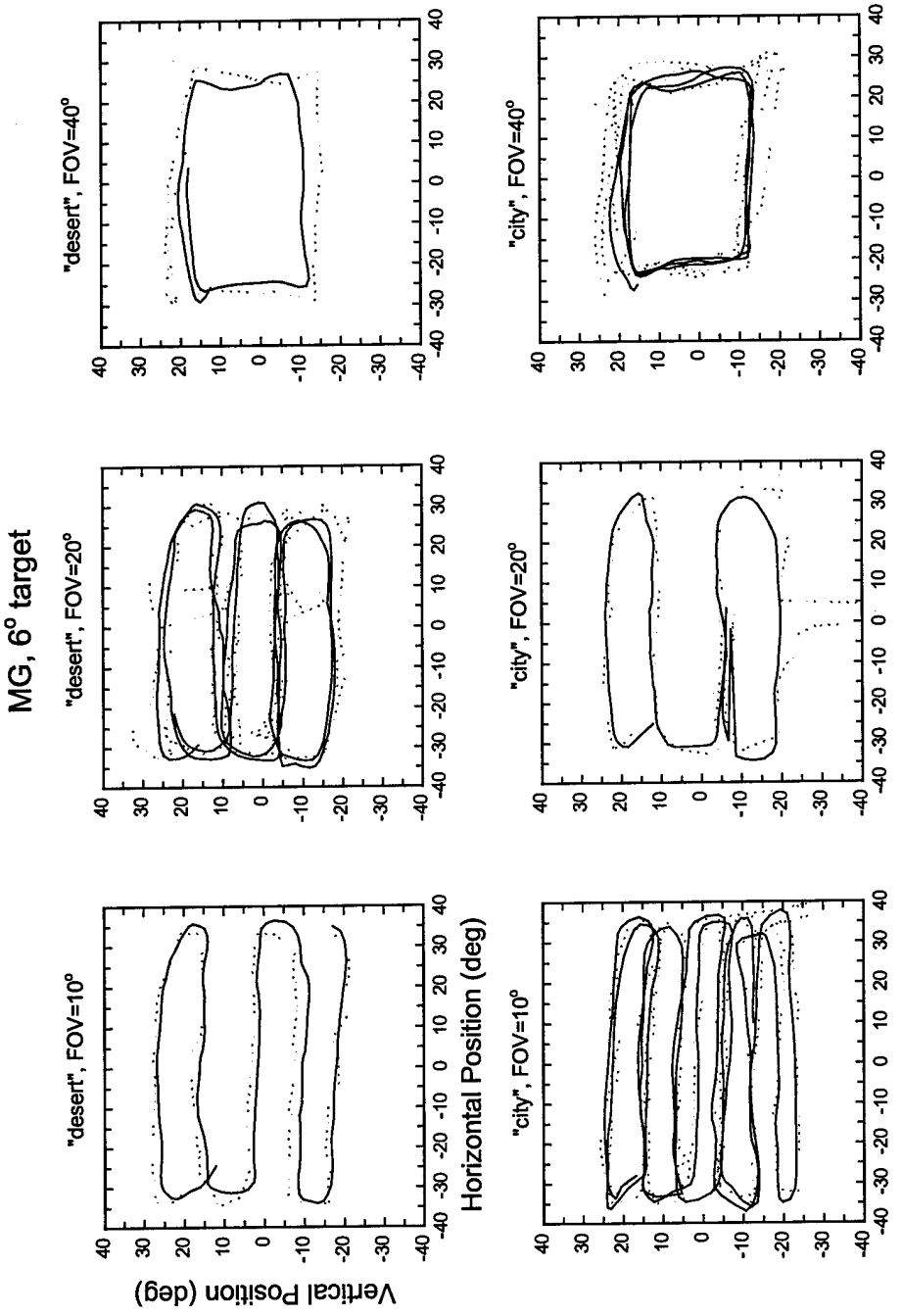

LM, 2° target
"desert", FOV=10°
"desert", FOV=20°
"desert", FOV=40°
"city", FOV=10°
"city", FOV=20°
"city", FOV=40°
Vertical Position (deg)
Horizontal Position (deg)

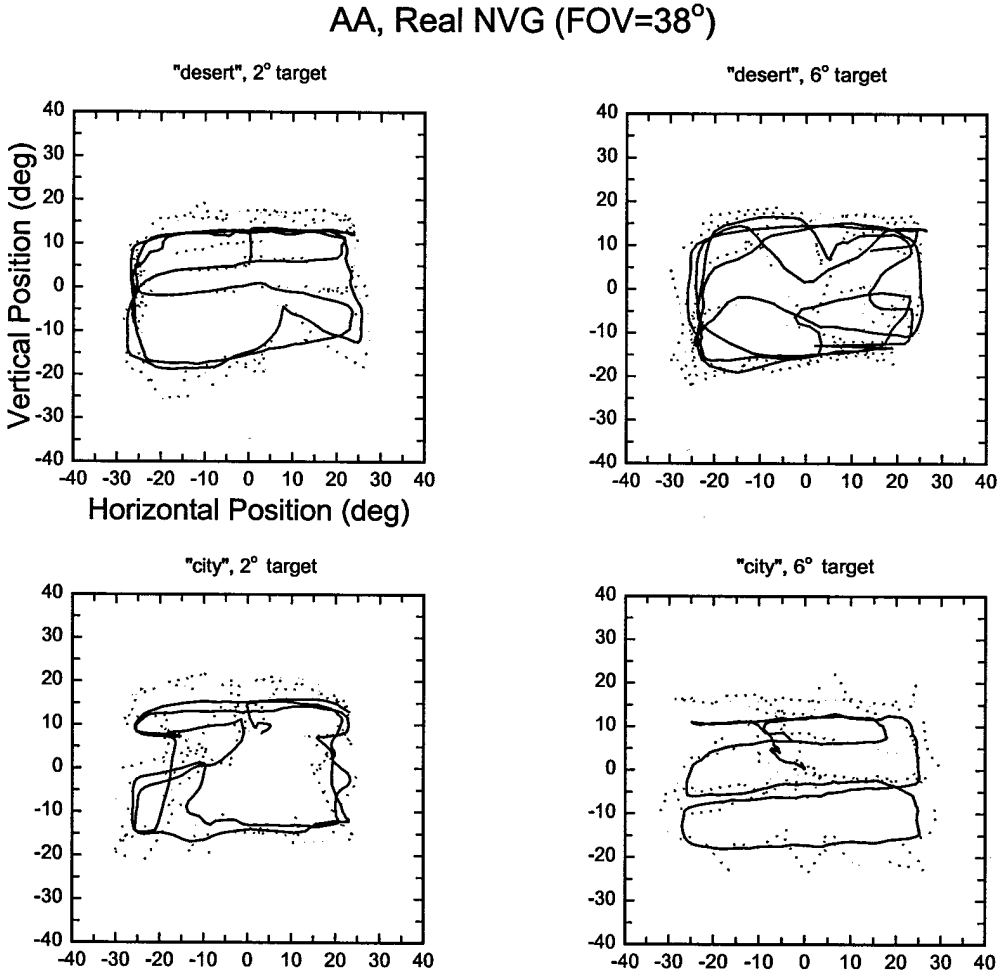

LM, 6° target
"desert", FOV=10°
"desert", FOV=20°
"desert", FOV=40°
"city", FOV=10°
"city", FOV=20°
"city", FOV=40°
Vertical Position (deg)
Horizontal Position (deg)
40
30
20
10
0
-10
-20
-30
-40
-40 -30 -20 -10 0 10 20 30 40

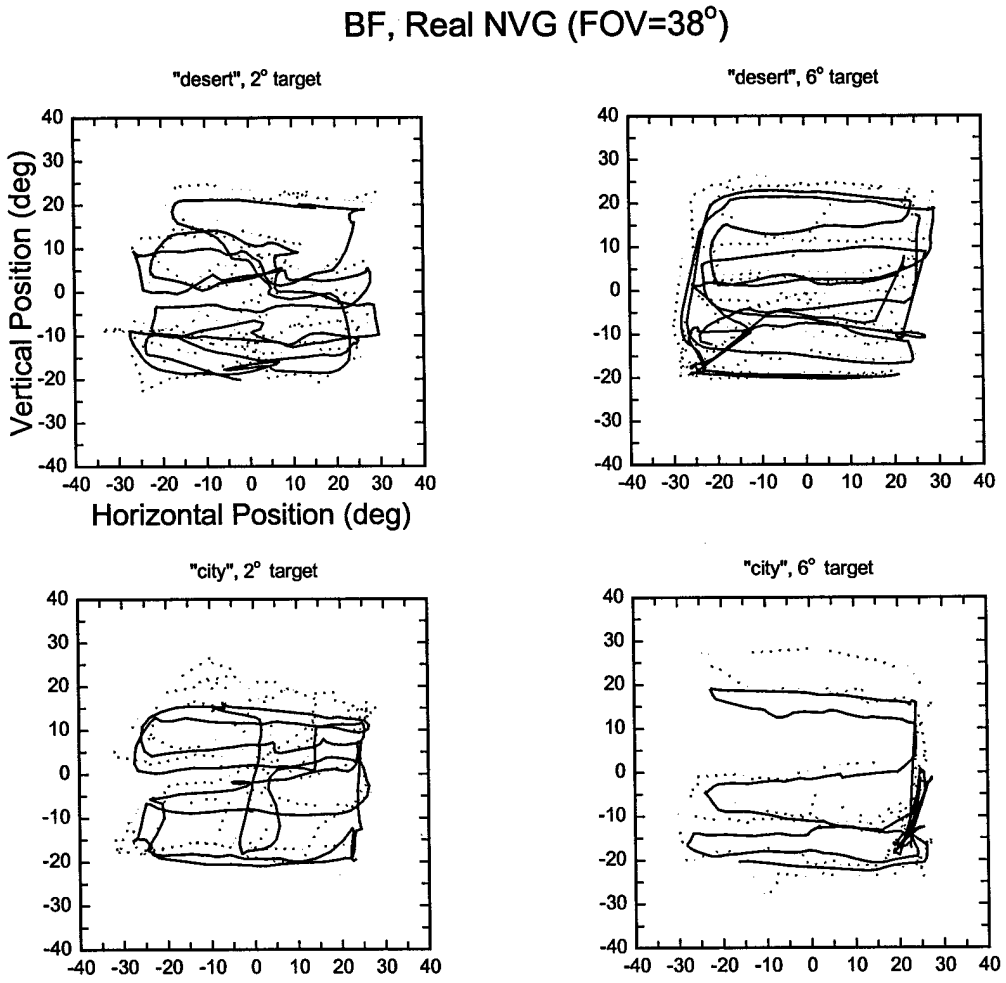

MC, 2° target
"desert", FOV=10°
"desert", FOV=20°
"desert", FOV=40°
"city", FOV=10°
"city", FOV=20°
"city", FOV=40°
Vertical Position (deg)
Horizontal Position (deg)
40
30
20
10
0
-10
-20
-30
-40
-40 -30 -20 -10 0 10 20 30 40

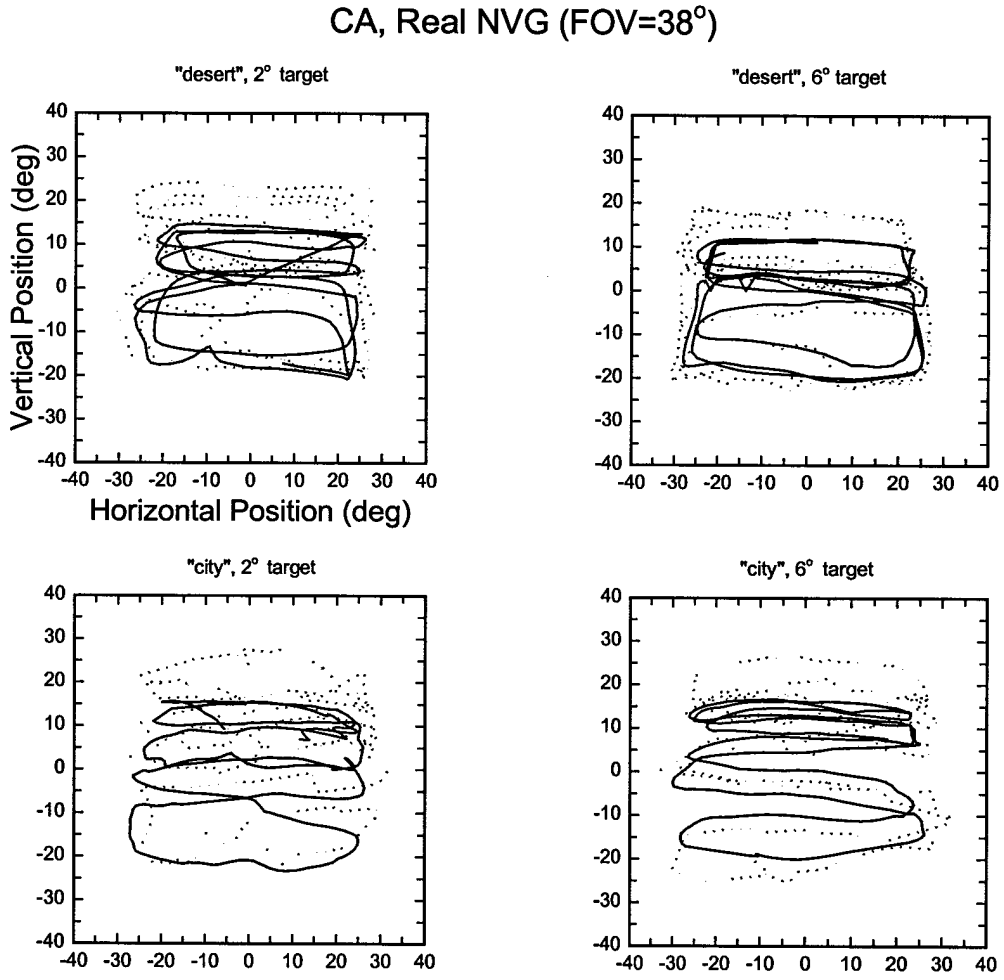

MC, 6° target
"desert", FOV=10°
"desert", FOV=20°
"desert", FOV=40°
"city", FOV=10°
"city", FOV=20°
"city", FOV=40°
Vertical Position (deg)
Horizontal Position (deg)
40
30
20
10
0
-10
-20
-30
-40
-40 -30 -20 -10 0 10 20 30 40

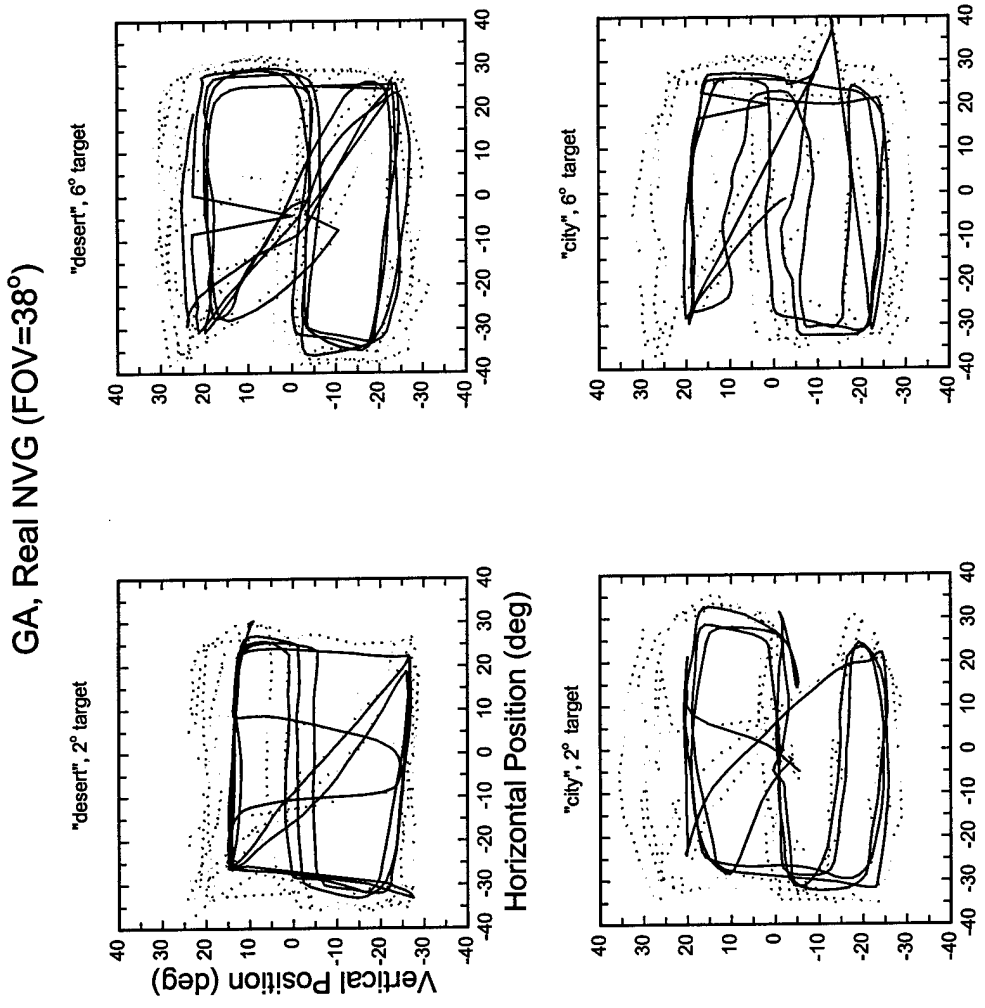

MG, 2° target
"desert", FOV=40°
"desert", FOV=20°
"desert", FOV=10°
"city", FOV=40°
"city", FOV=20°
"city", FOV=10°
Vertical Position (deg)
Horizontal Position (deg)

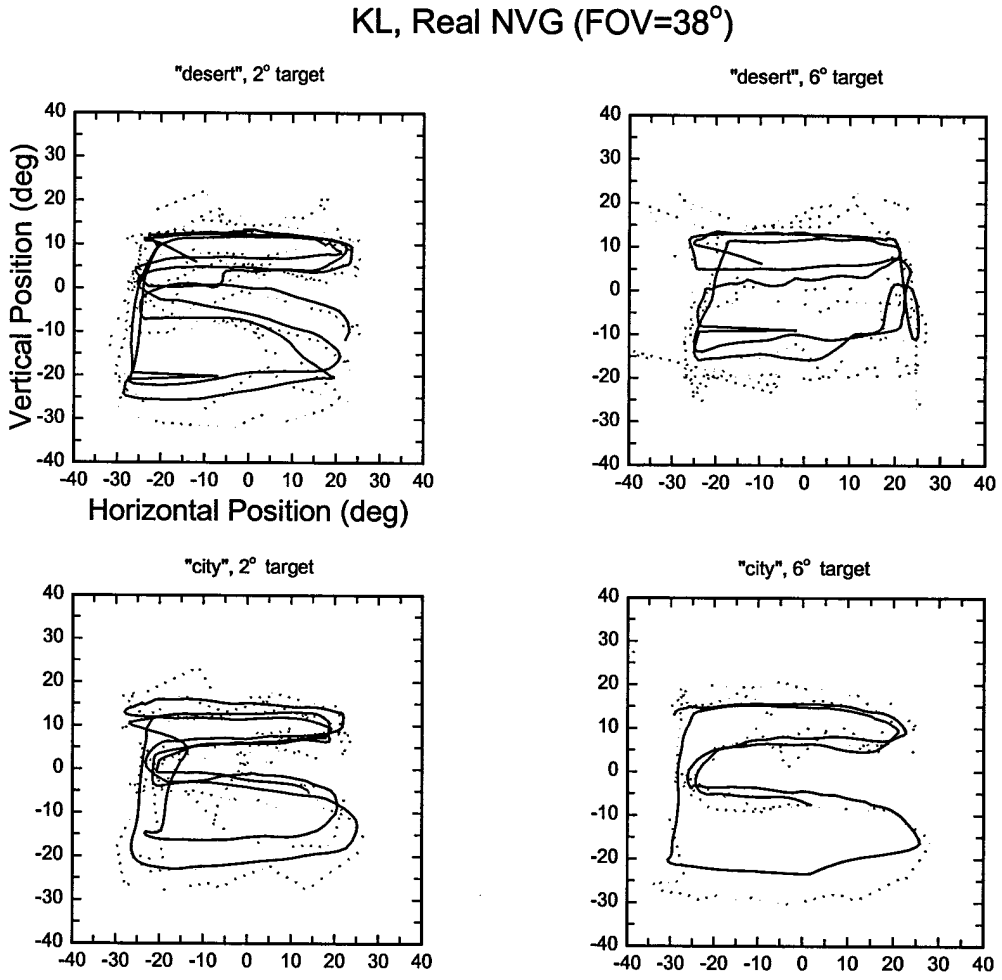

MG, 6° target
"desert", FOV=10°
"desert", FOV=20°
"desert", FOV=40°
"city", FOV=10°
"city", FOV=20°
"city", FOV=40°
Vertical Position (deg)
Horizontal Position (deg)


AA, Real NVG (FOV=38°)
"desert", 2° target
"desert", 6° target
"city", 2° target
"city", 6° target
Vertical Position (deg)
Horizontal Position (deg)
40
30
20
10
0
-10
-20
-30
-40


BF, Real NVG (FOV=38°)
"desert", 2° target
"desert", 6° target
"city", 2° target
"city", 6° target
Vertical Position (deg)
Horizontal Position (deg)
40
30
20
10
0
-10
-20
-30
-40

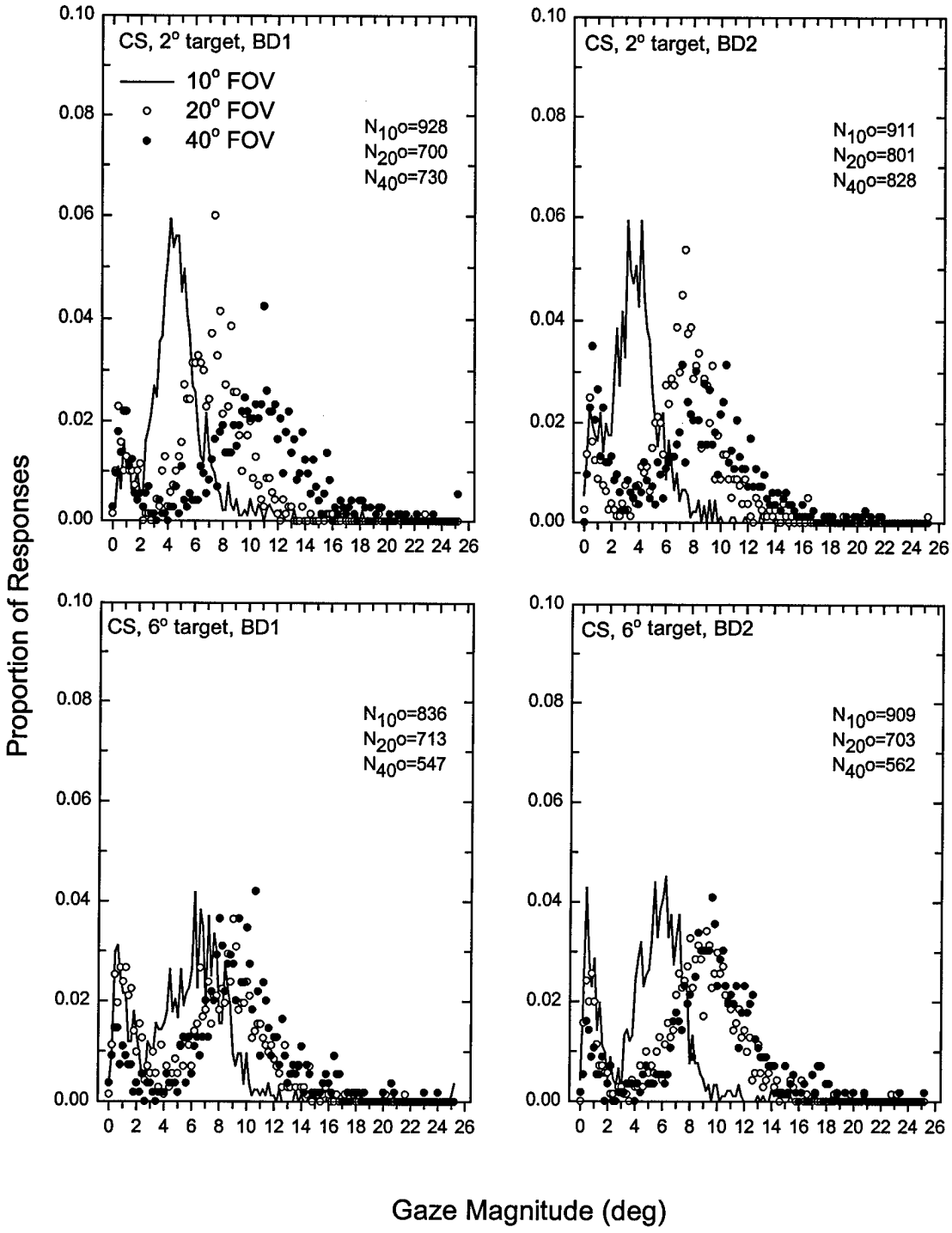

CA, Real NVG (FOV=38°)
"desert", 2° target
"desert", 6° target
"city", 2° target
"city", 6° target
Vertical Position (deg)
Horizontal Position (deg)
40
30
20
10
0
-10
-20
-30
-40

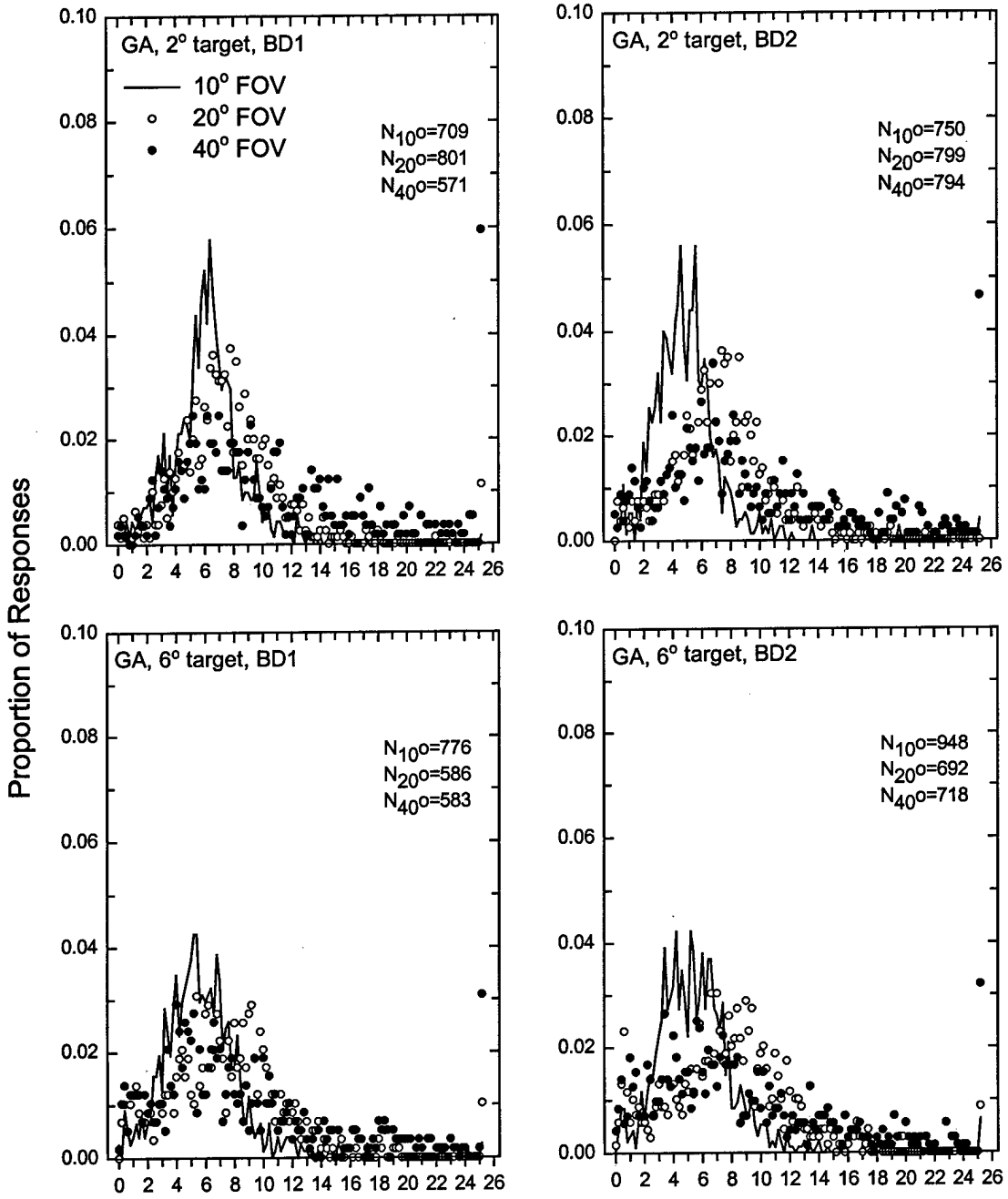

GA, Real NVG (FOV=38°)
"desert", 2° target
"desert", 6° target
"city", 2° target
"city", 6° target
Vertical Position (deg)
Horizontal Position (deg)

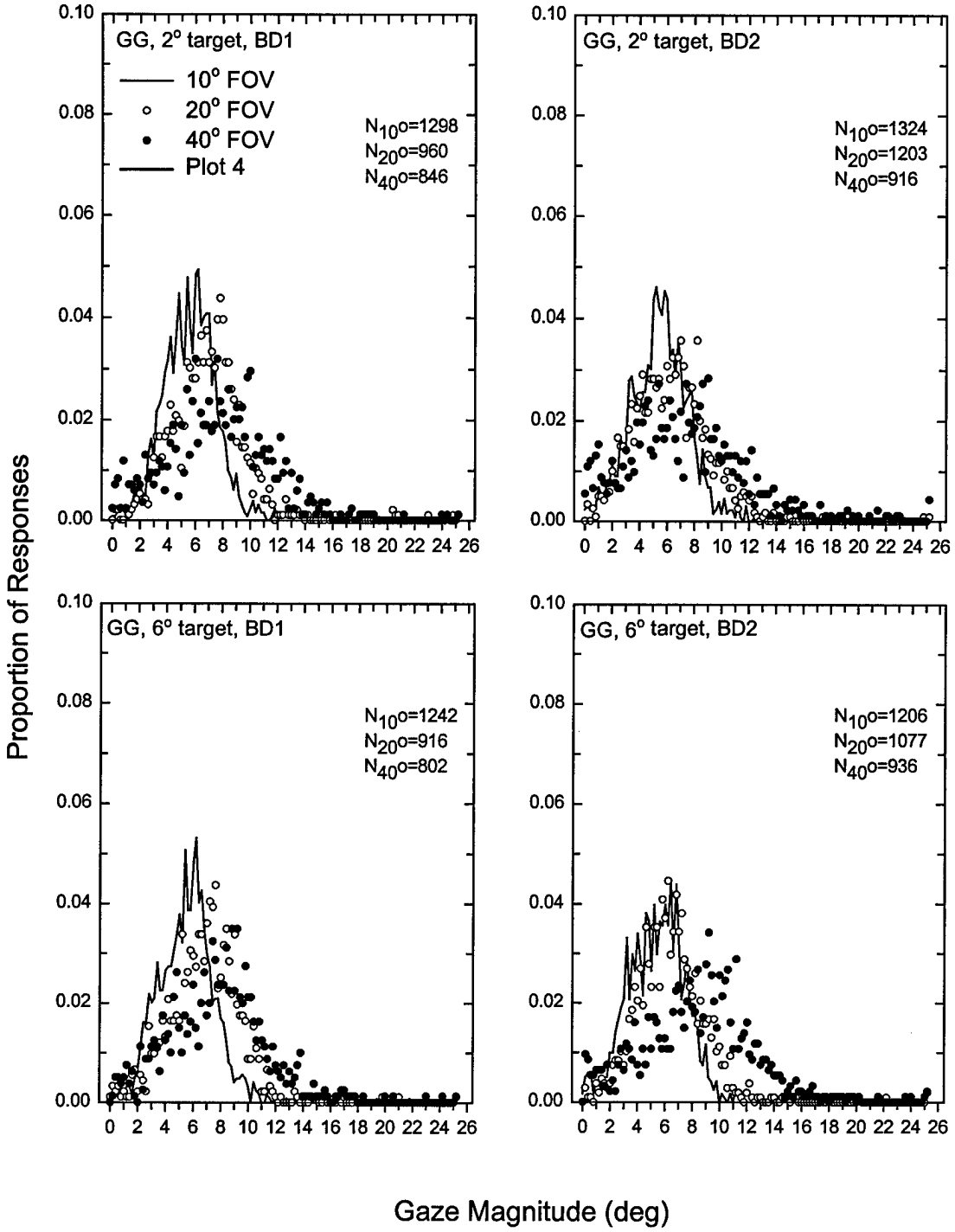

KL, Real NVG (FOV=38°)
"desert", 2° target
"desert", 6° target
"city", 2° target
"city", 6° target
Vertical Position (deg)
Horizontal Position (deg)
40
30
20
10
0
-10
-20
-30
-40

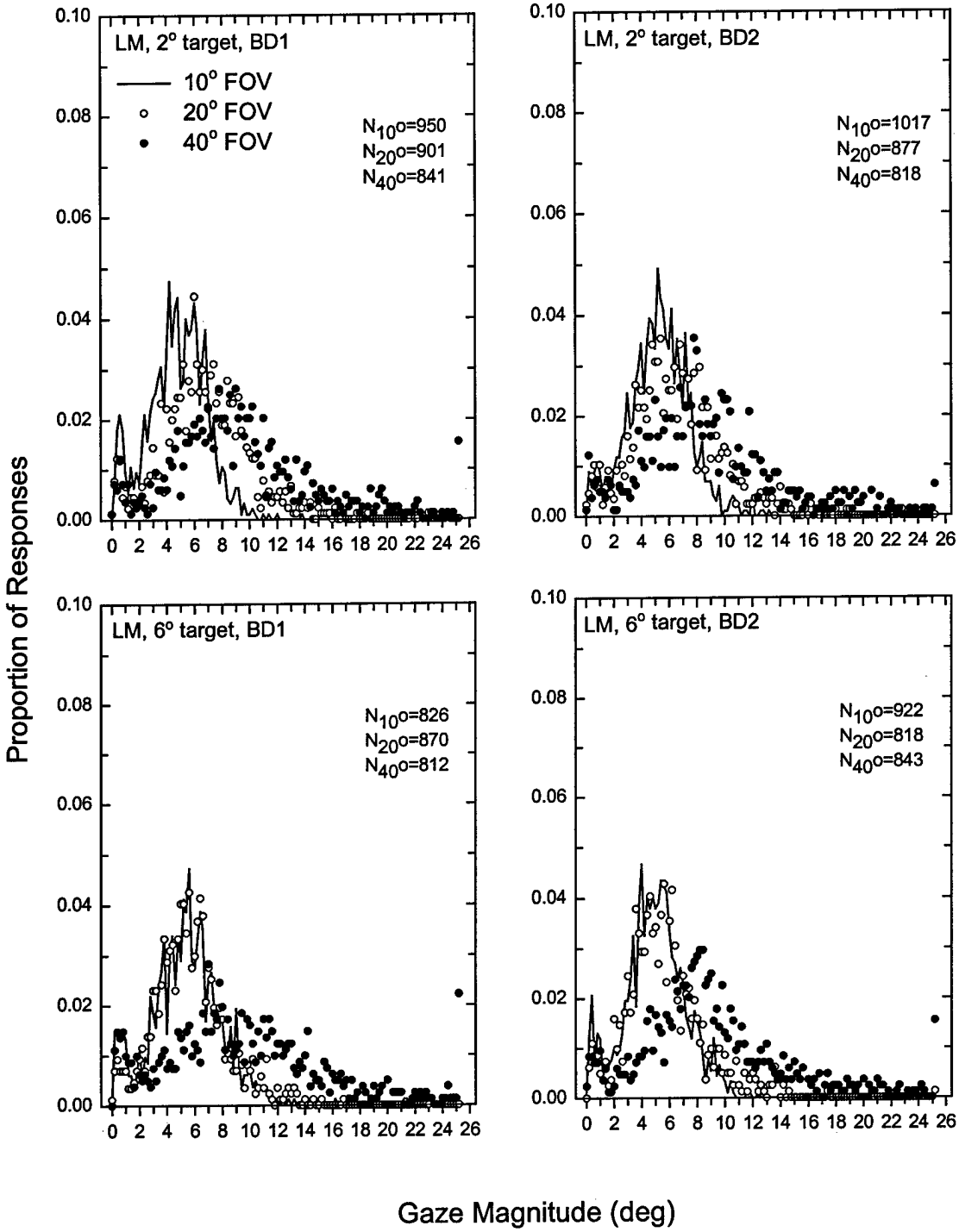
6.4 Appendix D

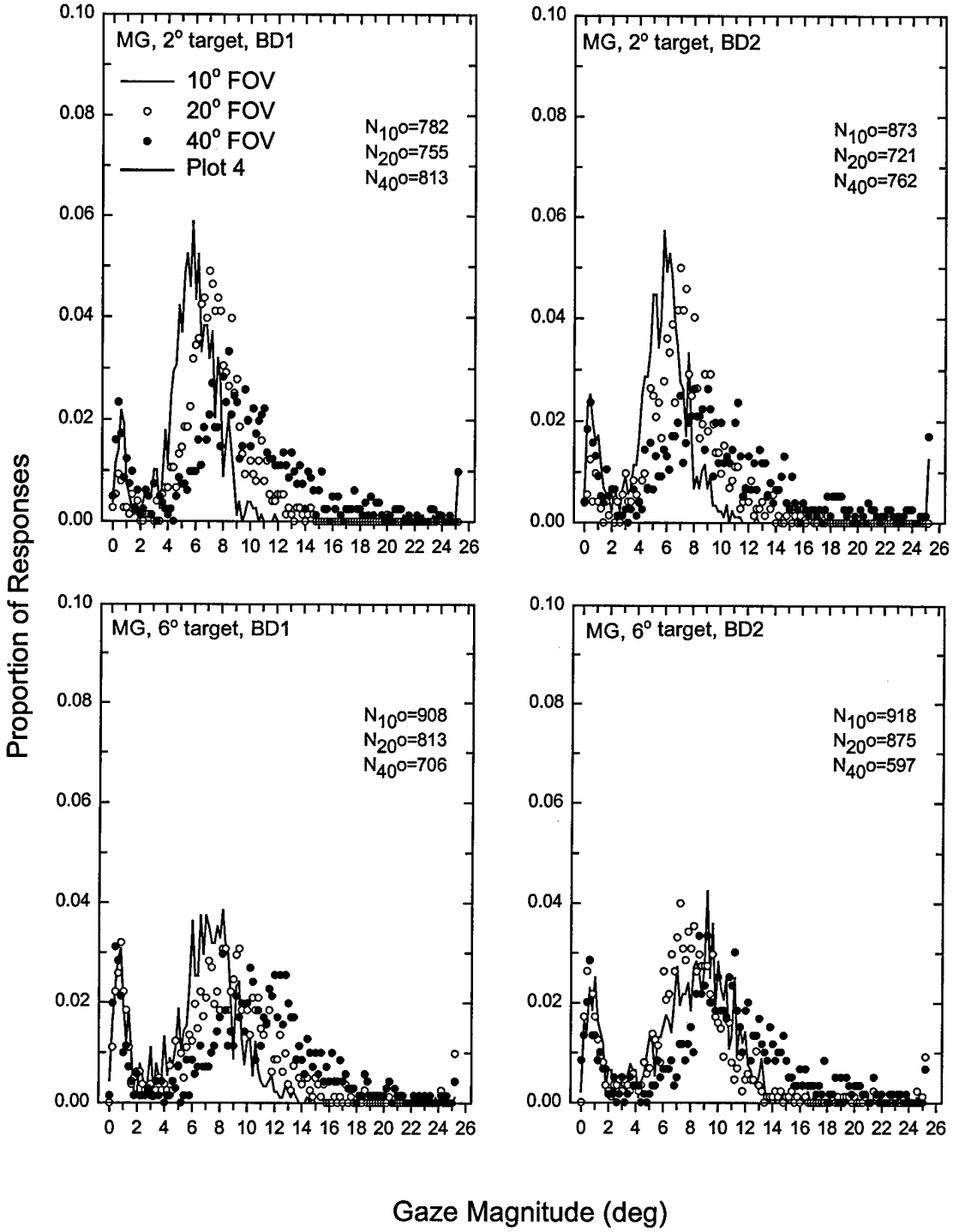
Gaze magnitude histograms obtained for all observers and all stimulus conditions tested under both the IFOV (three histograms per graph) and Real-NVG (one histogram per graph) conditions.

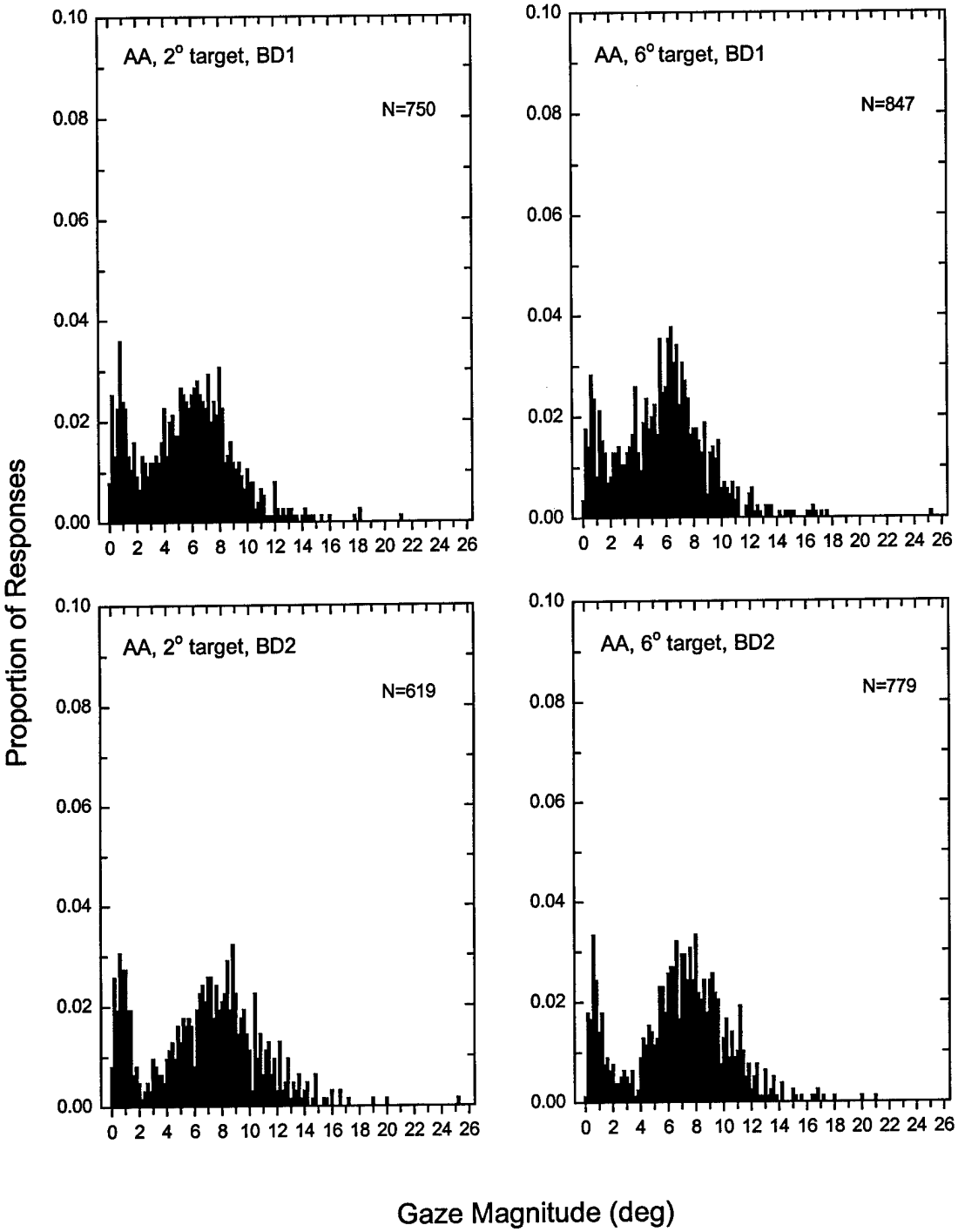

CS, 2° target, BD1
10° FOV
20° FOV
40° FOV
N10°=928
N20°=700
N40°=730
CS, 2° target, BD2
N10°=911
N20°=801
N40°=828
CS, 6° target, BD1
N10°=836
N20°=713
N40°=547
CS, 6° target, BD2
N10°=909
N20°=703
N40°=562
Proportion of Responses
Gaze Magnitude (deg)

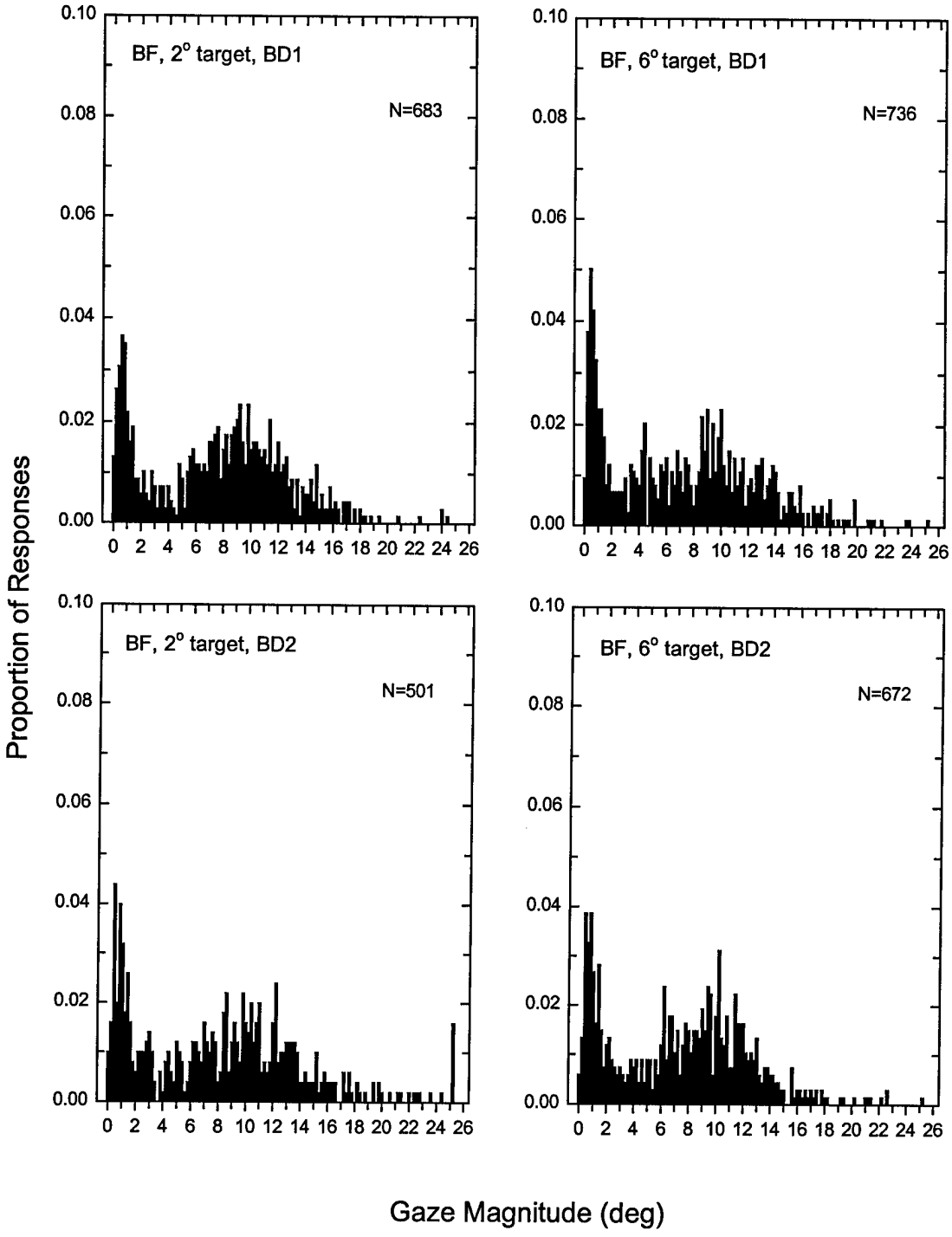

GA, 2° target, BD1
10° FOV
20° FOV
40° FOV
N10°=709
N20°=801
N40°=571
GA, 2° target, BD2
N10°=750
N20°=799
N40°=794
GA, 6° target, BD1
N10°=776
N20°=586
N40°=583
GA, 6° target, BD2
N10°=948
N20°=692
N40°=718
Proportion of Responses
Gaze Magnitude (deg)

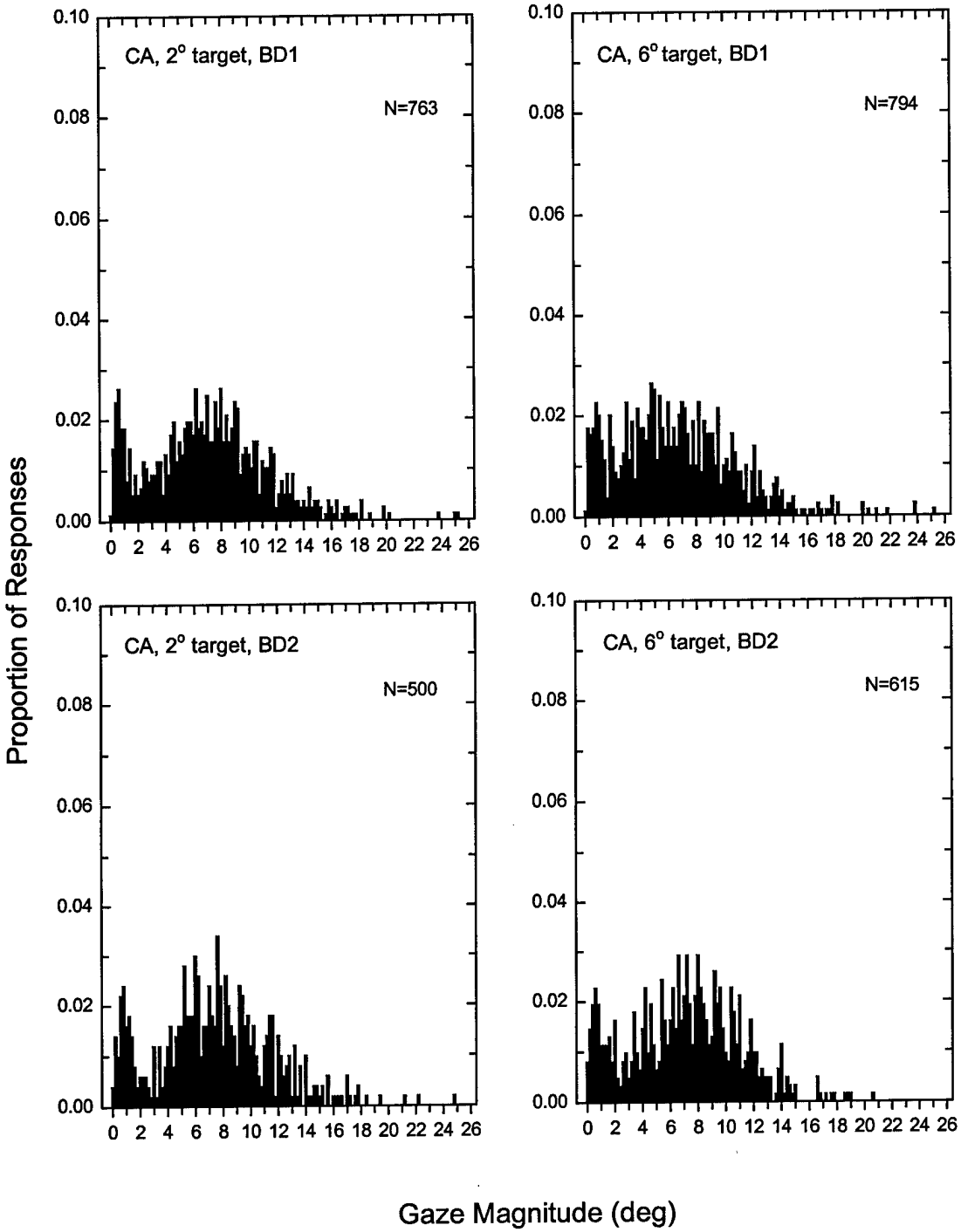

GG, 2° target, BD1
10° FOV
20° FOV
40° FOV
Plot 4
N10o=1298
N20o=960
N40o=846
GG, 2° target, BD2
N10o=1324
N20o=1203
N40o=916
GG, 6° target, BD1
N10o=1242
N20o=916
N40o=802
GG, 6° target, BD2
N10o=1206
N20o=1077
N40o=936
Proportion of Responses
Gaze Magnitude (deg)

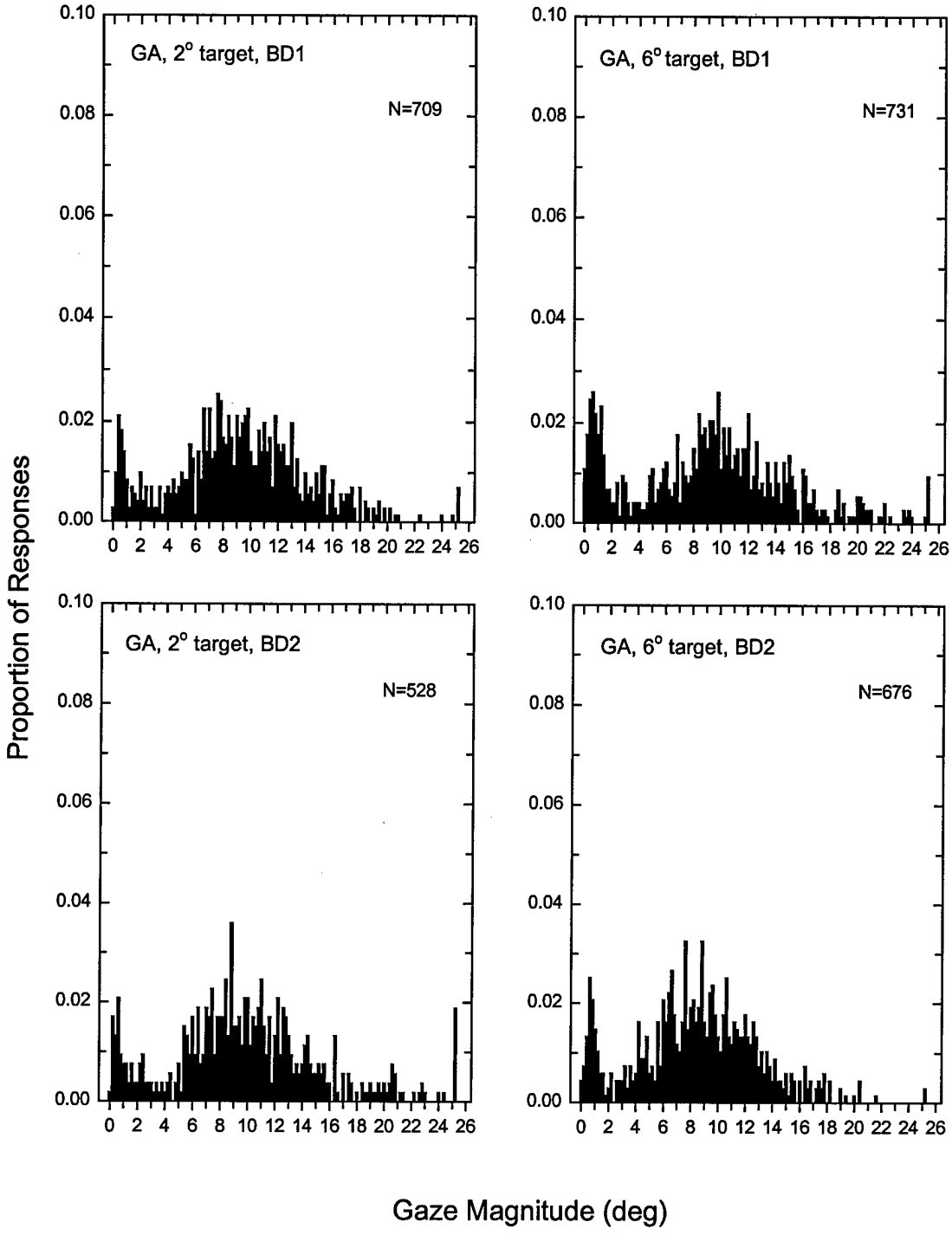

LM, 2° target, BD1
10° FOV
20° FOV
40° FOV
N10°=950
N20°=901
N40°=841
LM, 2° target, BD2
N10°=1017
N20°=877
N40°=818
LM, 6° target, BD1
N10°=826
N20°=870
N40°=812
LM, 6° target, BD2
N10°=922
N20°=818
N40°=843
Proportion of Responses
Gaze Magnitude (deg)

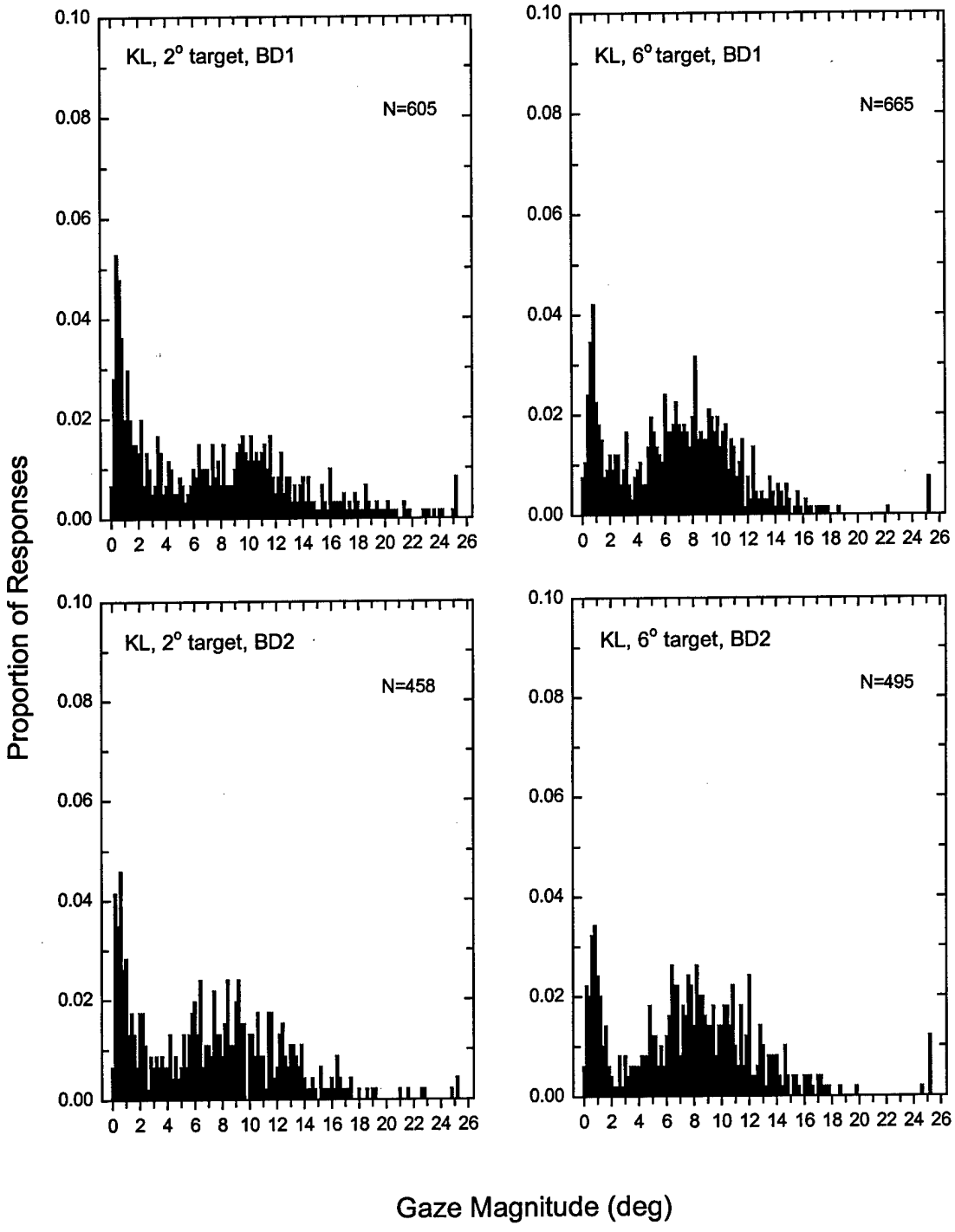

MG, 2° target, BD1
10° FOV
20° FOV
40° FOV
Plot 4
N10°=782
N20°=755
N40°=813
MG, 2° target, BD2
N10°=873
N20°=721
N40°=762
MG, 6° target, BD1
N10°=908
N20°=813
N40°=706
MG, 6° target, BD2
N10°=918
N20°=875
N40°=597
Proportion of Responses
Gaze Magnitude (deg)


AA, 2° target, BD1
N=750
AA, 6° target, BD1
N=847
AA, 2° target, BD2
N=619
AA, 6° target, BD2
N=779
Proportion of Responses
Gaze Magnitude (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0 2 4 6 8 10 12 14 16 18 20 22 24 26


BF, 2° target, BD1
N=683
BF, 6° target, BD1
N=736
BF, 2° target, BD2
N=501
BF, 6° target, BD2
N=672
Proportion of Responses
Gaze Magnitude (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0 2 4 6 8 10 12 14 16 18 20 22 24 26


CA, 2° target, BD1
N=763
CA, 6° target, BD1
N=794
CA, 2° target, BD2
N=500
CA, 6° target, BD2
N=615
Proportion of Responses
Gaze Magnitude (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0 2 4 6 8 10 12 14 16 18 20 22 24 26

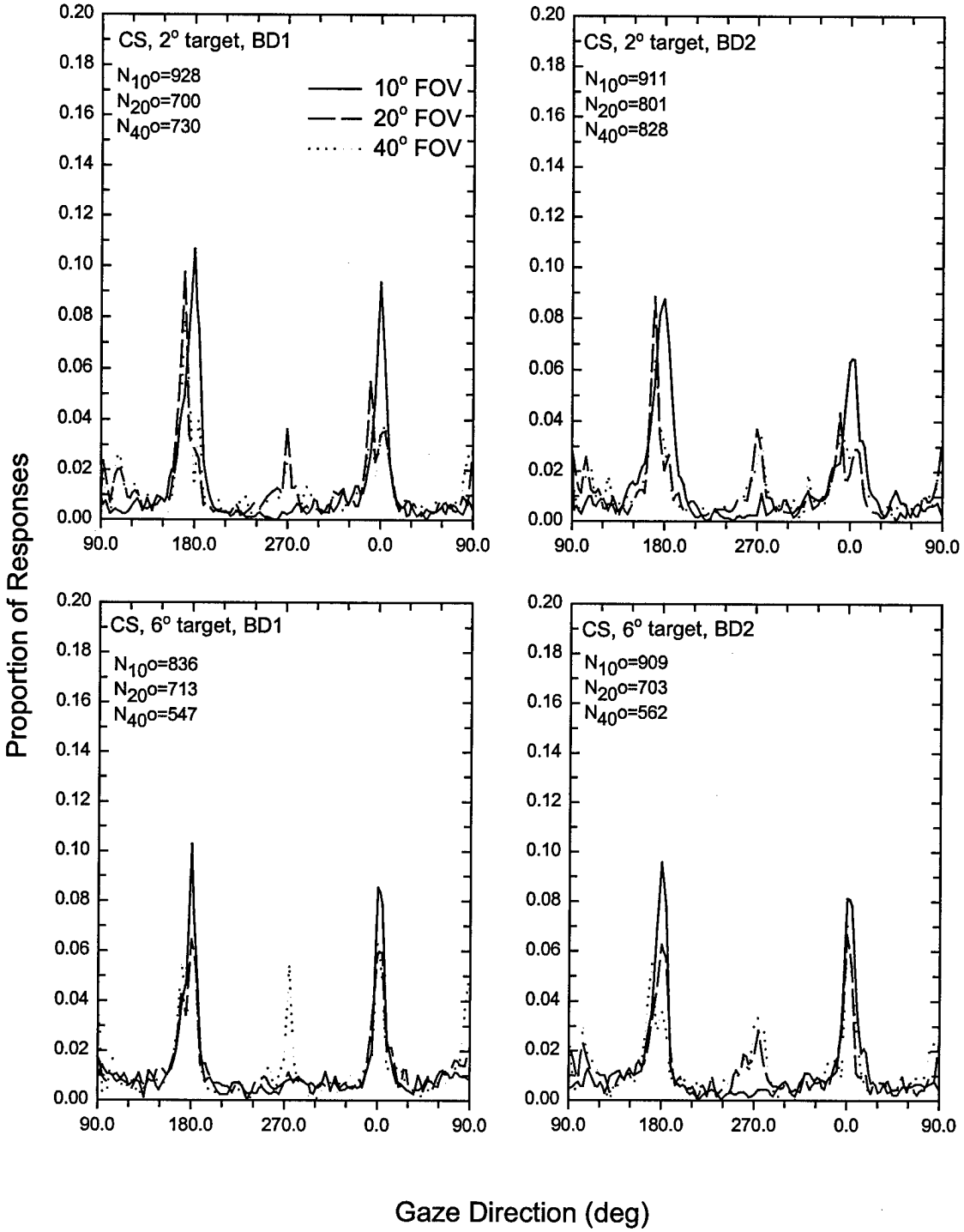

GA, 2° target, BD1
N=709
GA, 6° target, BD1
N=731
GA, 2° target, BD2
N=528
GA, 6° target, BD2
N=676
Proportion of Responses
Gaze Magnitude (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0 2 4 6 8 10 12 14 16 18 20 22 24 26

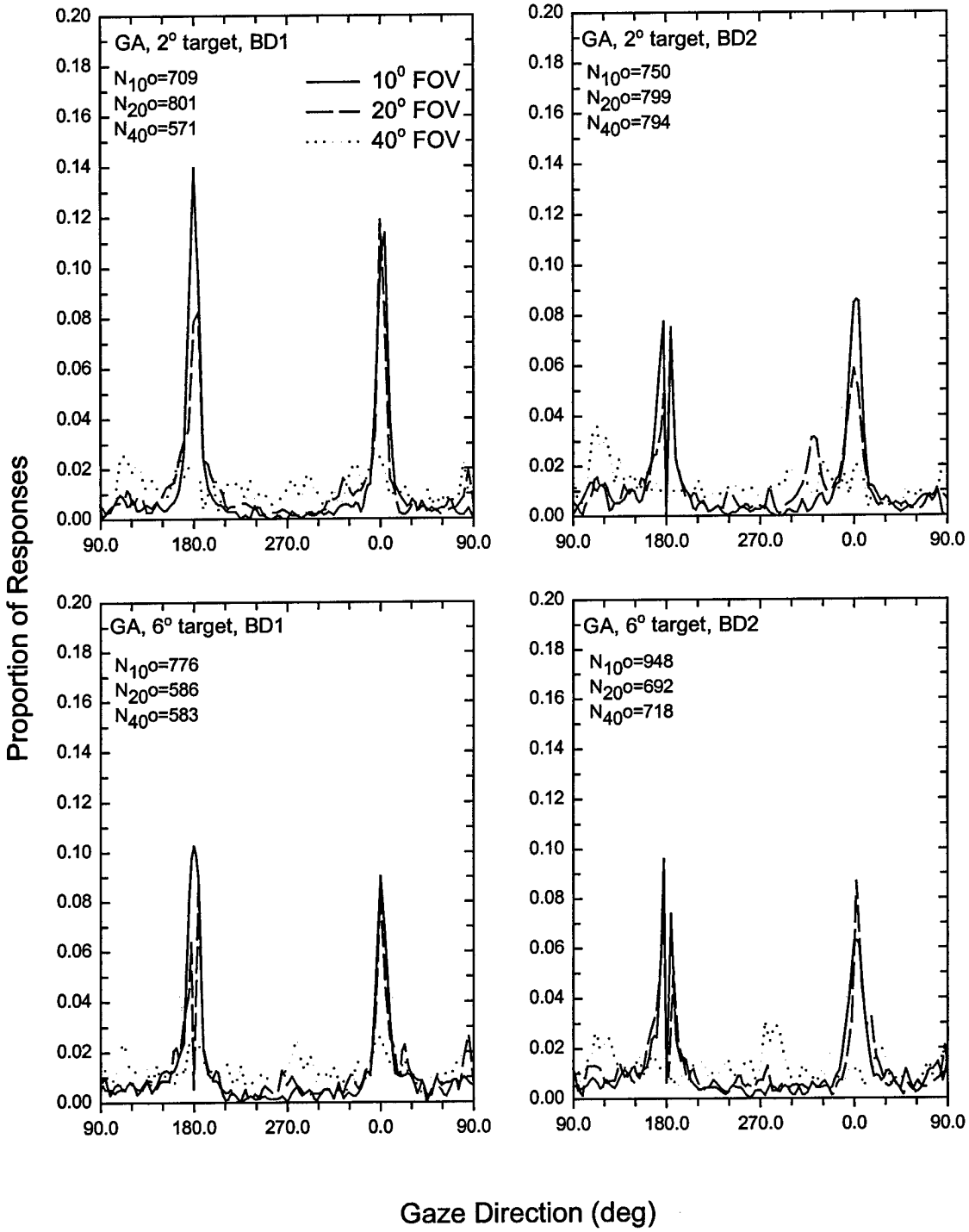

KL, 2° target, BD1
N=605
KL, 6° target, BD1
N=665
KL, 2° target, BD2
N=458
KL, 6° target, BD2
N=495
Proportion of Responses
Gaze Magnitude (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0 2 4 6 8 10 12 14 16 18 20 22 24 26

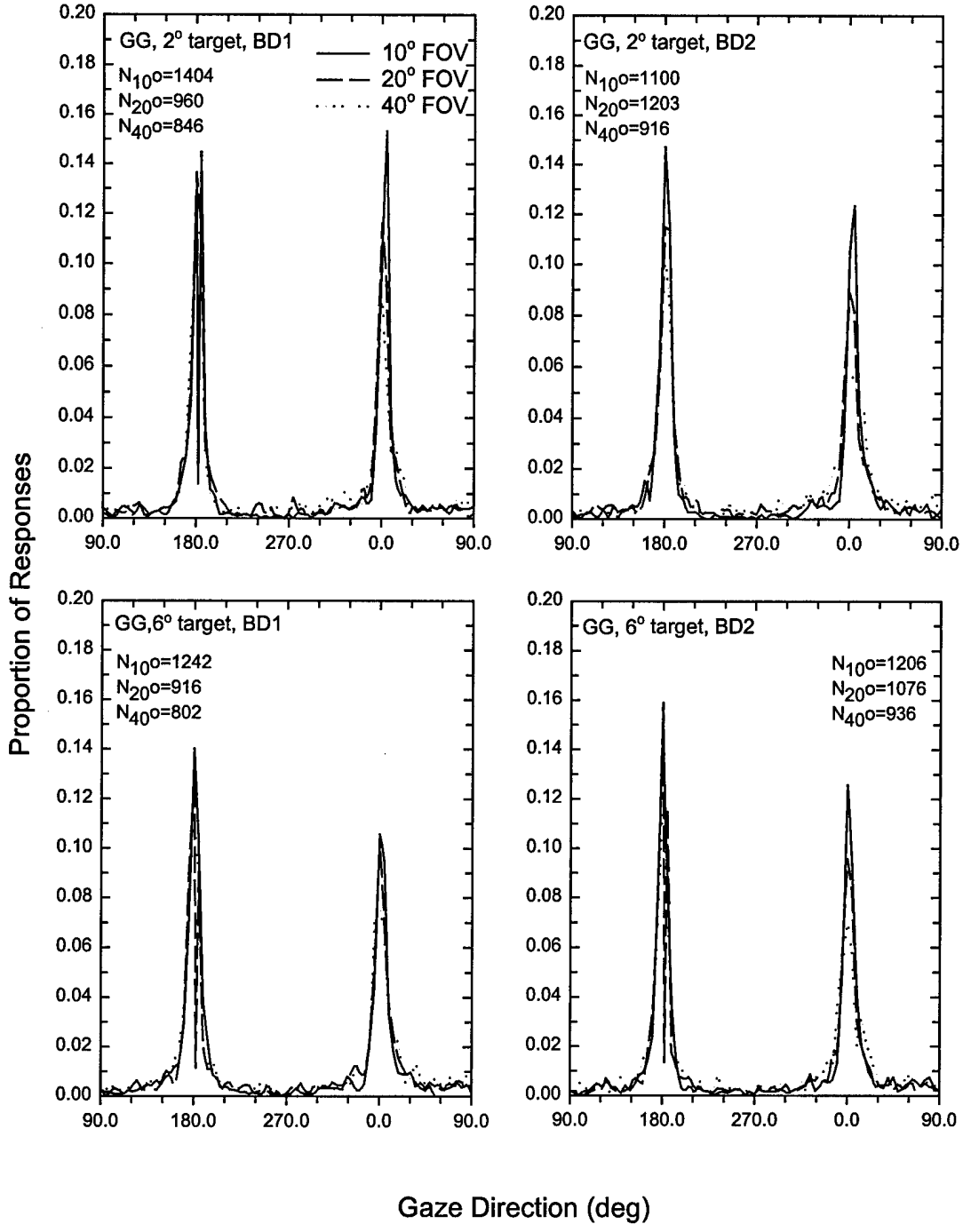
This page intentionally left blank.

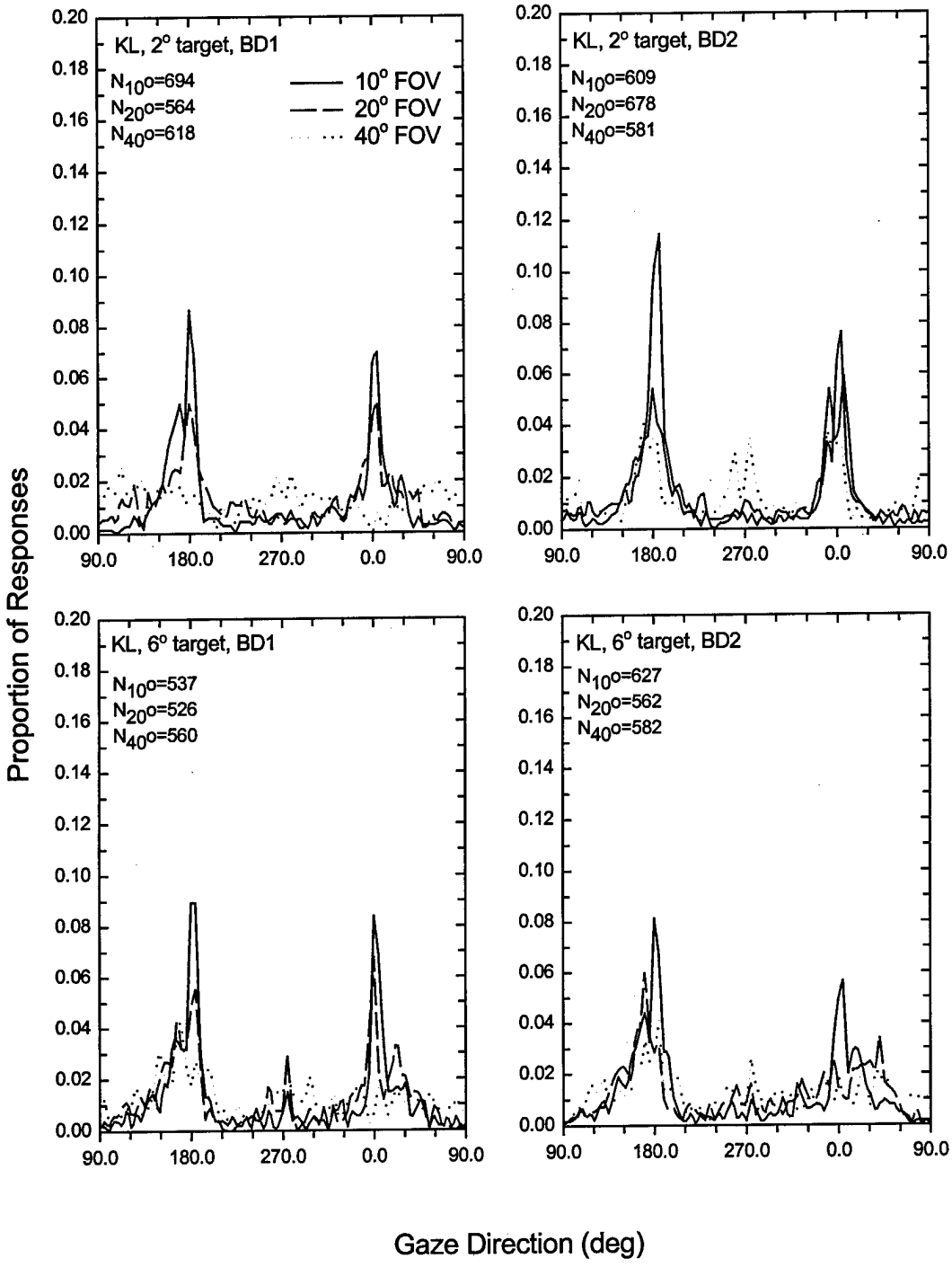
6.5 Appendix E

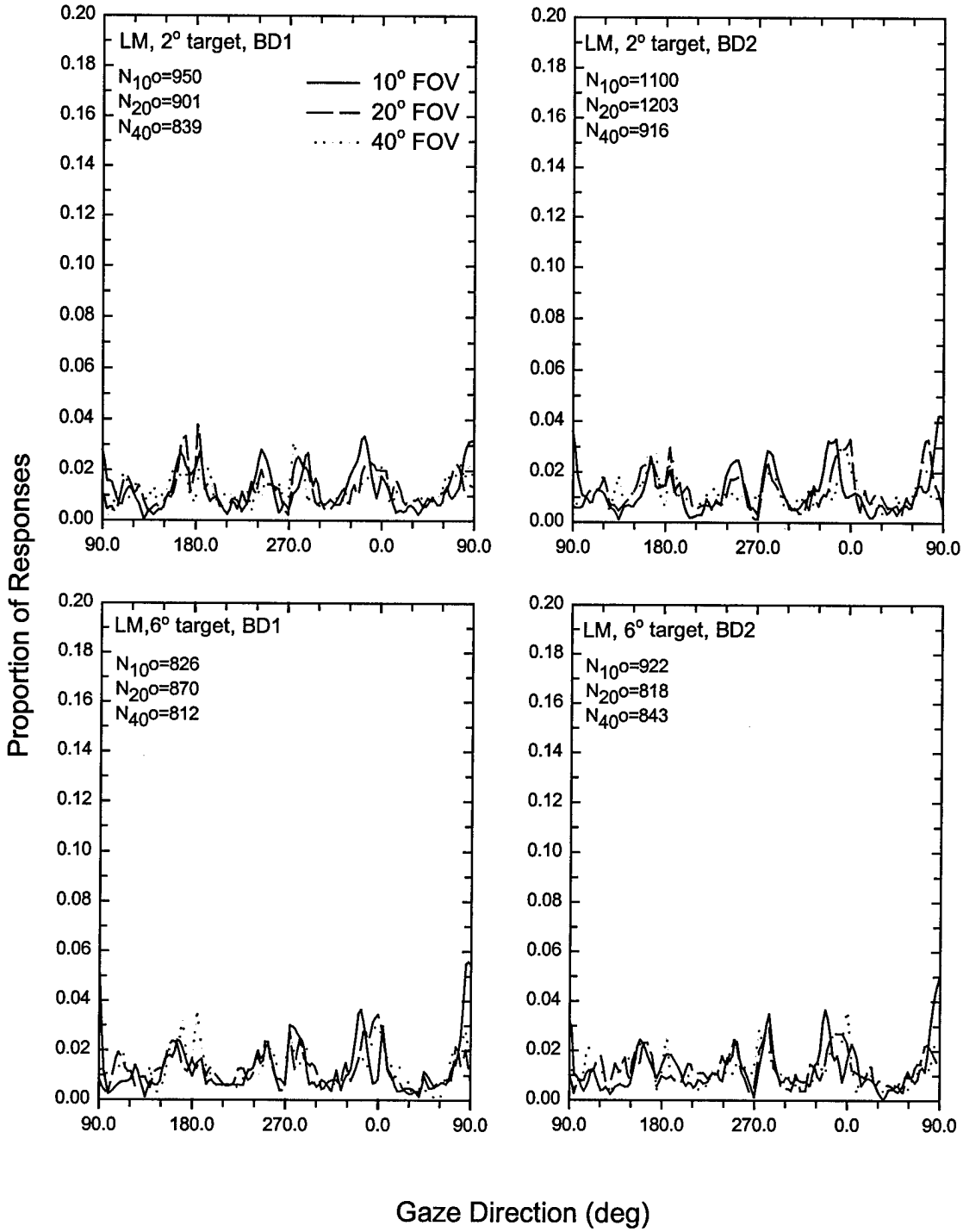
Gaze direction histograms obtained for all observers and all stimulus conditions tested under both the IFOV (three histograms per graph) and Real-NVG (one histogram per graph) conditions.

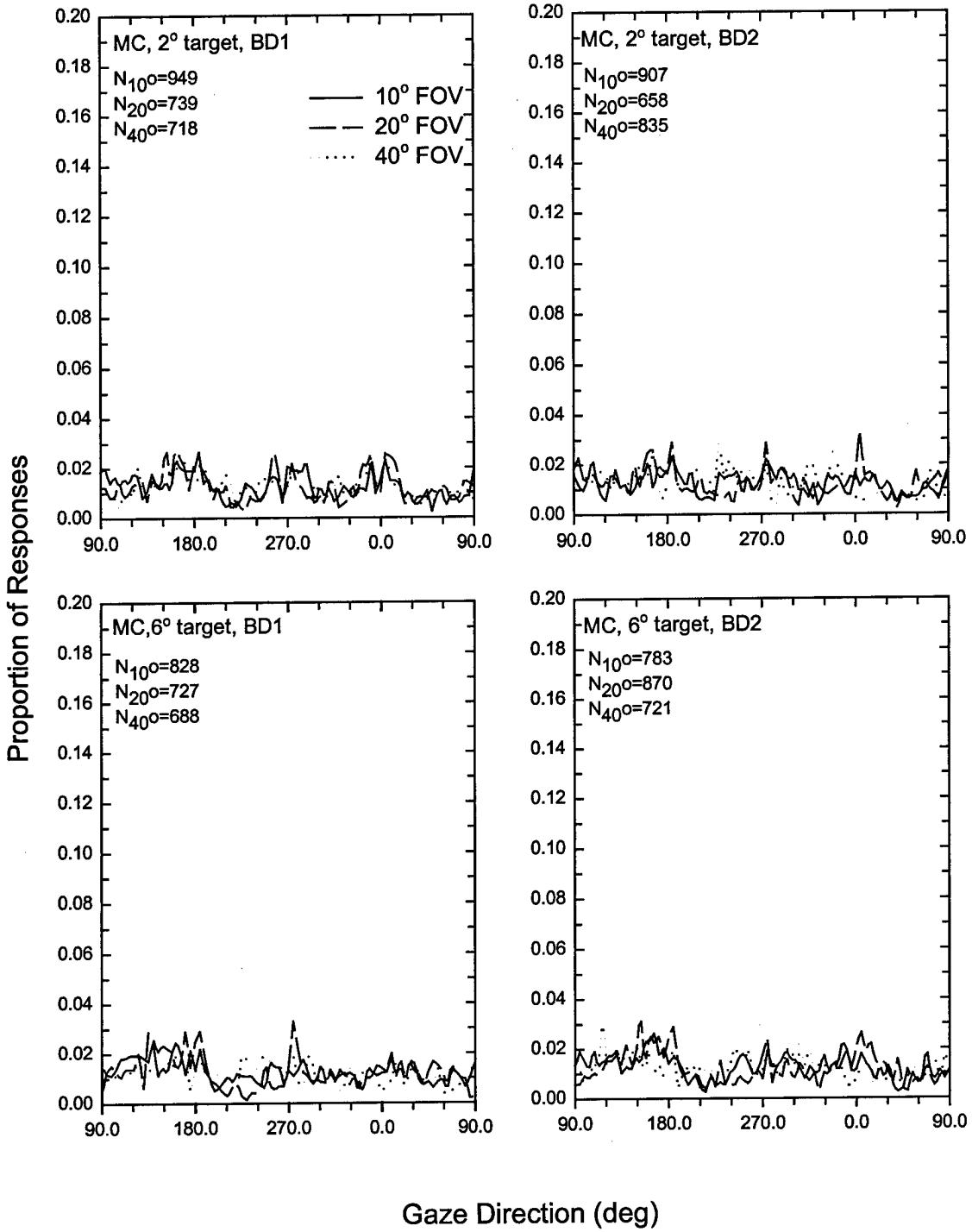

CS, 2° target, BD1
N10°=928
N20°=700
N40°=730
10° FOV
20° FOV
40° FOV
CS, 2° target, BD2
N10°=911
N20°=801
N40°=828
CS, 6° target, BD1
N10°=836
N20°=713
N40°=547
CS, 6° target, BD2
N10°=909
N20°=703
N40°=562
Proportion of Responses
Gaze Direction (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.20
90.0
180.0
270.0
0.0
90.0

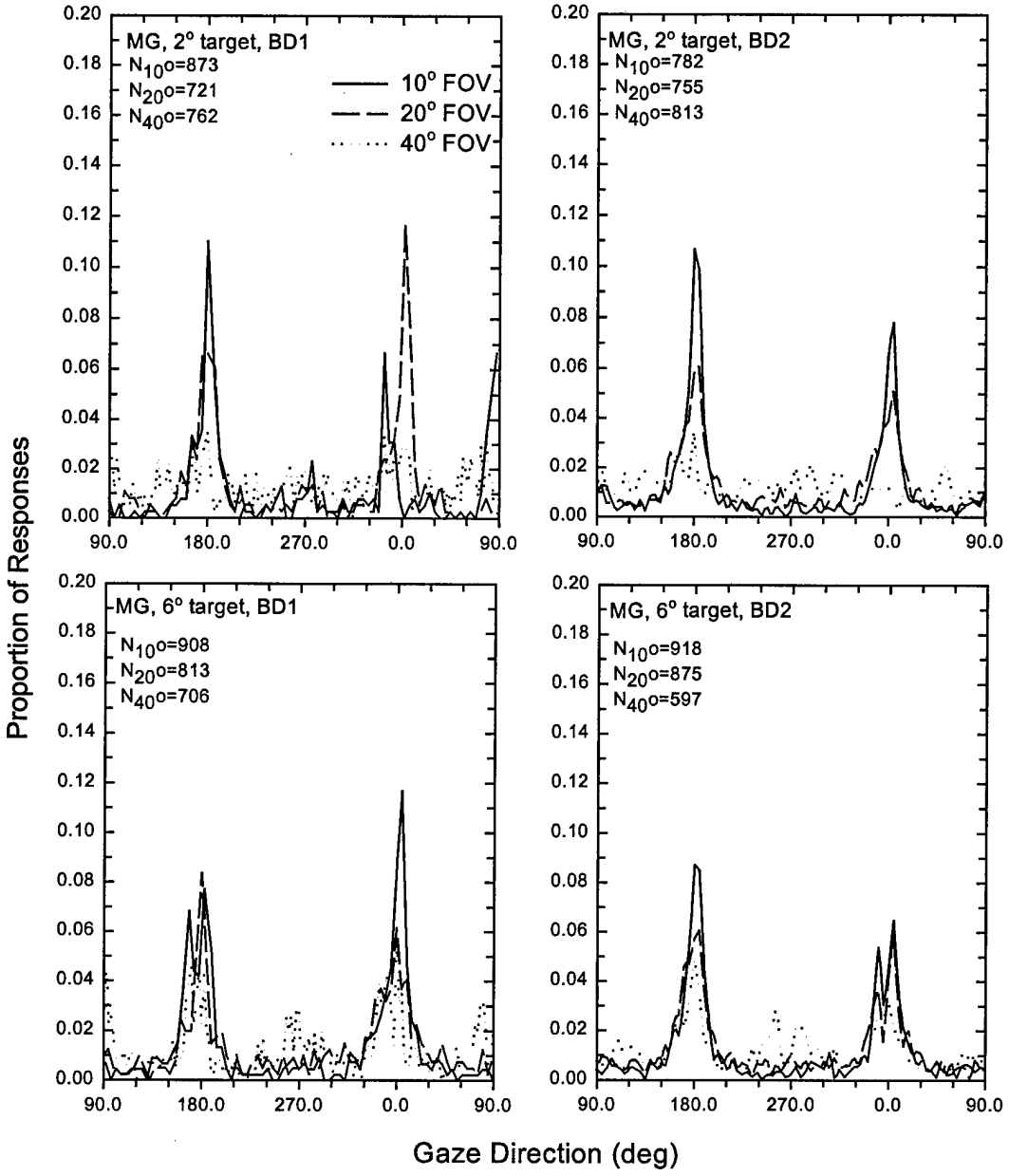

GA, 2° target, BD1
N10°=709
N20°=801
N40°=571
10° FOV
20° FOV
40° FOV
GA, 2° target, BD2
N10°=750
N20°=799
N40°=794
GA, 6° target, BD1
N10°=776
N20°=586
N40°=583
GA, 6° target, BD2
N10°=948
N20°=692
N40°=718
Proportion of Responses
Gaze Direction (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.20
90.0
180.0
270.0
0.0
90.0

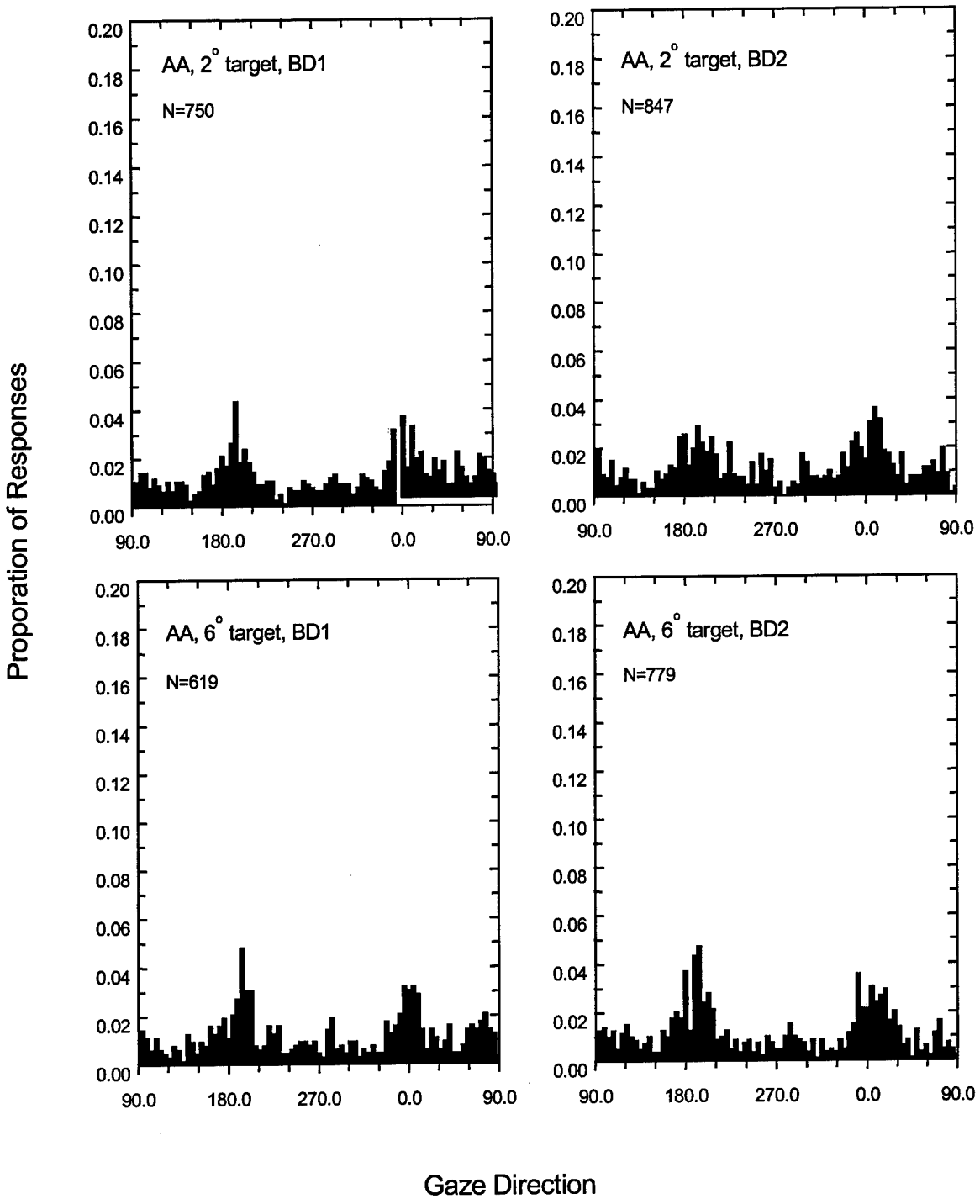

GG, 2° target, BD1
10° FOV
20° FOV
40° FOV
N10°=1404
N20°=960
N40°=846
GG, 2° target, BD2
N10°=1100
N20°=1203
N40°=916
GG,6° target, BD1
N10°=1242
N20°=916
N40°=802
GG, 6° target, BD2
N10°=1206
N20°=1076
N40°=936
Proportion of Responses
Gaze Direction (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.20
90.0
180.0
270.0
0.0
90.0

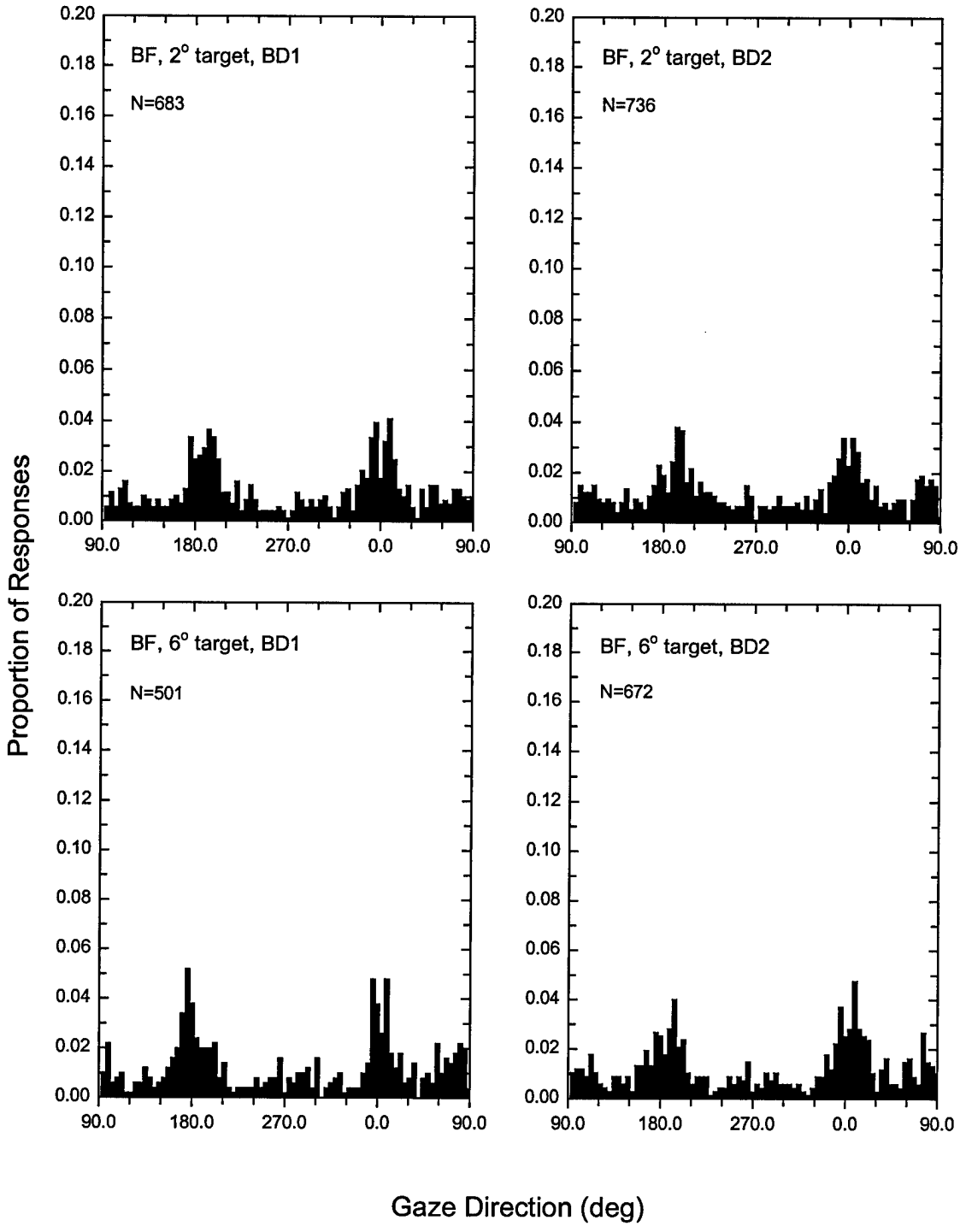

KL, 2° target, BD1
N10°=694
N20°=564
N40°=618
10° FOV
20° FOV
40° FOV
KL, 2° target, BD2
N10°=609
N20°=678
N40°=581
KL, 6° target, BD1
N10°=537
N20°=526
N40°=560
KL, 6° target, BD2
N10°=627
N20°=562
N40°=582
Proportion of Responses
Gaze Direction (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.20
90.0
180.0
270.0
0.0
90.0

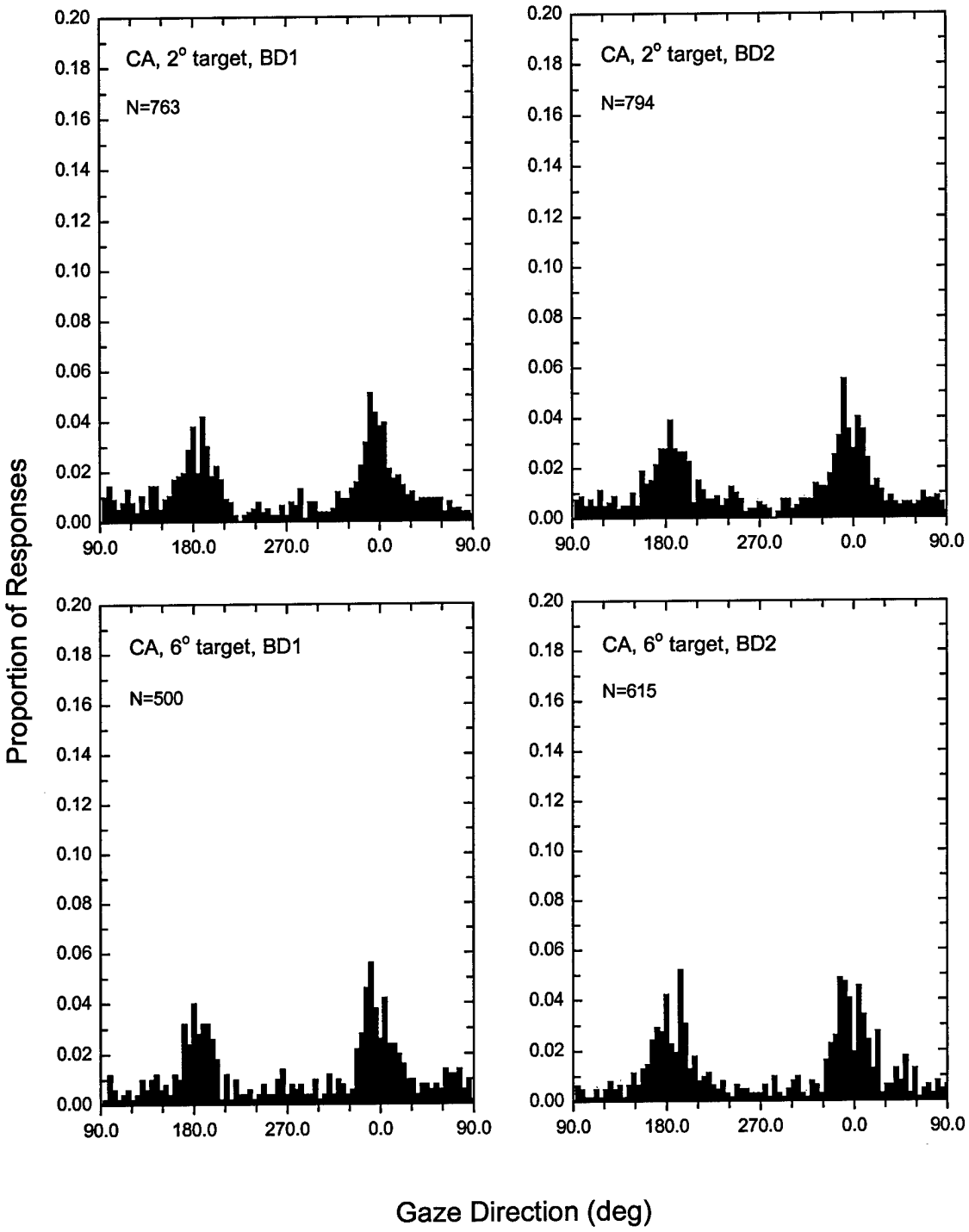

LM, 2° target, BD1
N10°=950
N20°=901
N40°=839
10° FOV
20° FOV
40° FOV
LM, 2° target, BD2
N10°=1100
N20°=1203
N40°=916
LM,6° target, BD1
N10°=826
N20°=870
N40°=812
LM, 6° target, BD2
N10°=922
N20°=818
N40°=843
Proportion of Responses
Gaze Direction (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.20
90.0
180.0
270.0
0.0
90.0

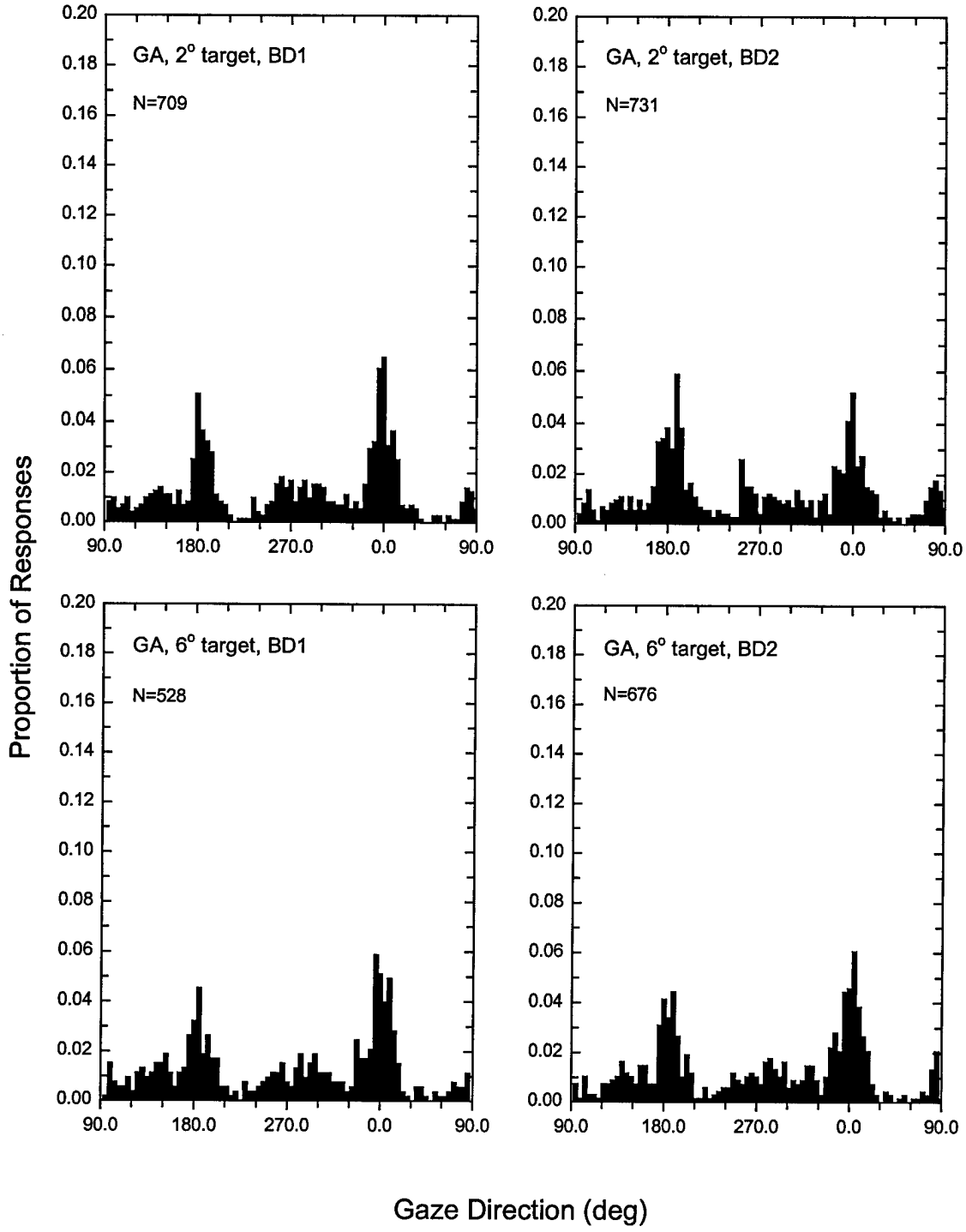

MC, 2° target, BD1
N10°=949
N20°=739
N40°=718
10° FOV
20° FOV
40° FOV
MC, 2° target, BD2
N10°=907
N20°=658
N40°=835
MC,6° target, BD1
N10°=828
N20°=727
N40°=688
MC, 6° target, BD2
N10°=783
N20°=870
N40°=721
Proportion of Responses
Gaze Direction (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.20
90.0
180.0
270.0
0.0
90.0

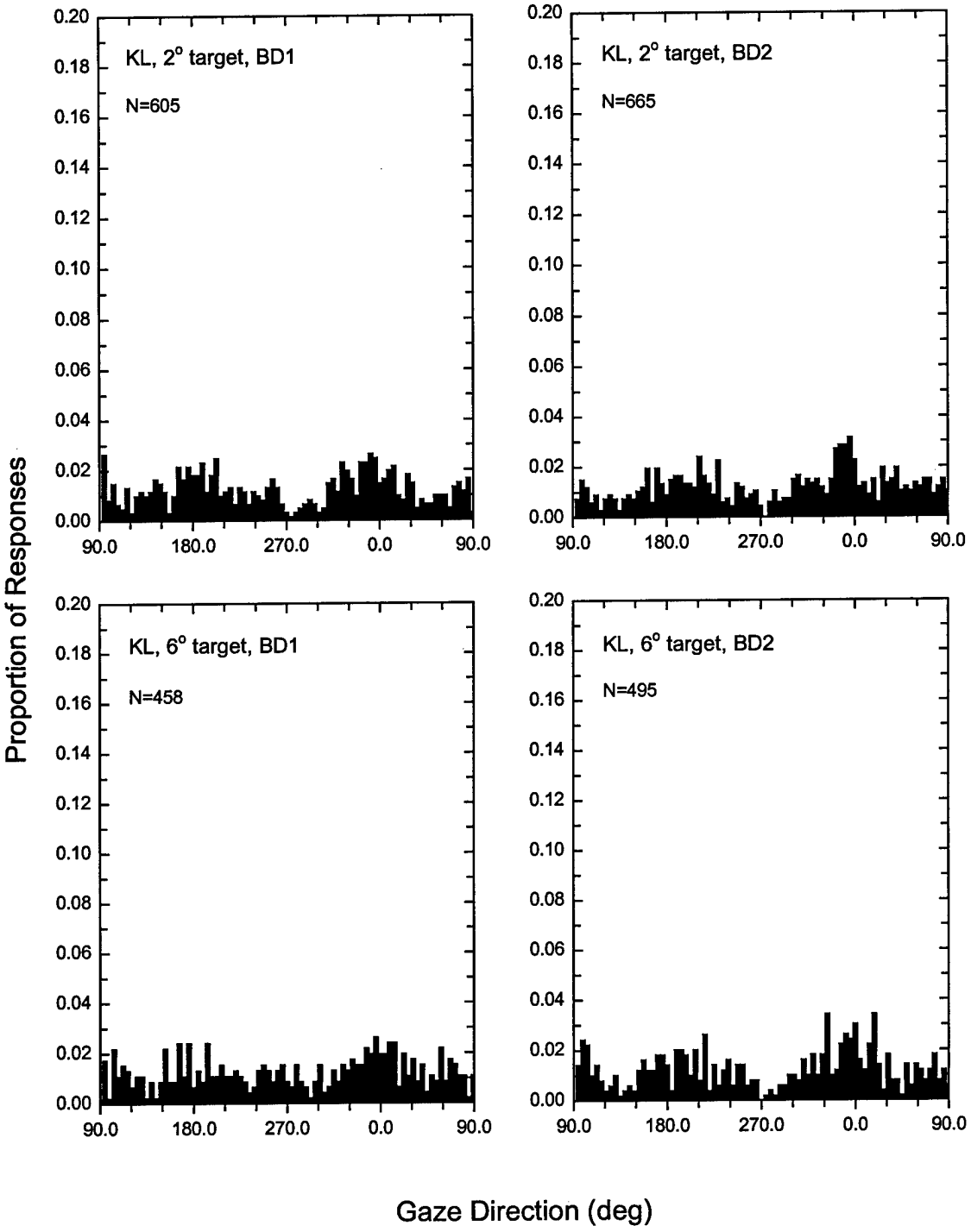

MG, 2° target, BD1
N10°=873
N20°=721
N40°=762
10° FOV
20° FOV
40° FOV
MG, 2° target, BD2
N10°=782
N20°=755
N40°=813
MG, 6° target, BD1
N10°=908
N20°=813
N40°=706
MG, 6° target, BD2
N10°=918
N20°=875
N40°=597
Proportion of Responses
Gaze Direction (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.20
90.0
180.0
270.0
0.0
90.0


AA, 2° target, BD1
N=750
AA, 2° target, BD2
N=847
AA, 6° target, BD1
N=619
AA, 6° target, BD2
N=779
Proporation of Responses
Gaze Direction
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.20
90.0
180.0
270.0
0.0
90.0


BF, 2° target, BD1
N=683
BF, 2° target, BD2
N=736
BF, 6° target, BD1
N=501
BF, 6° target, BD2
N=672
Proportion of Responses
Gaze Direction (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.20
90.0
180.0
270.0
0.0
90.0


CA, 2° target, BD1
N=763
CA, 2° target, BD2
N=794
CA, 6° target, BD1
N=500
CA, 6° target, BD2
N=615
Proportion of Responses
Gaze Direction (deg)
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
90.0
180.0
270.0
0.0
90.0

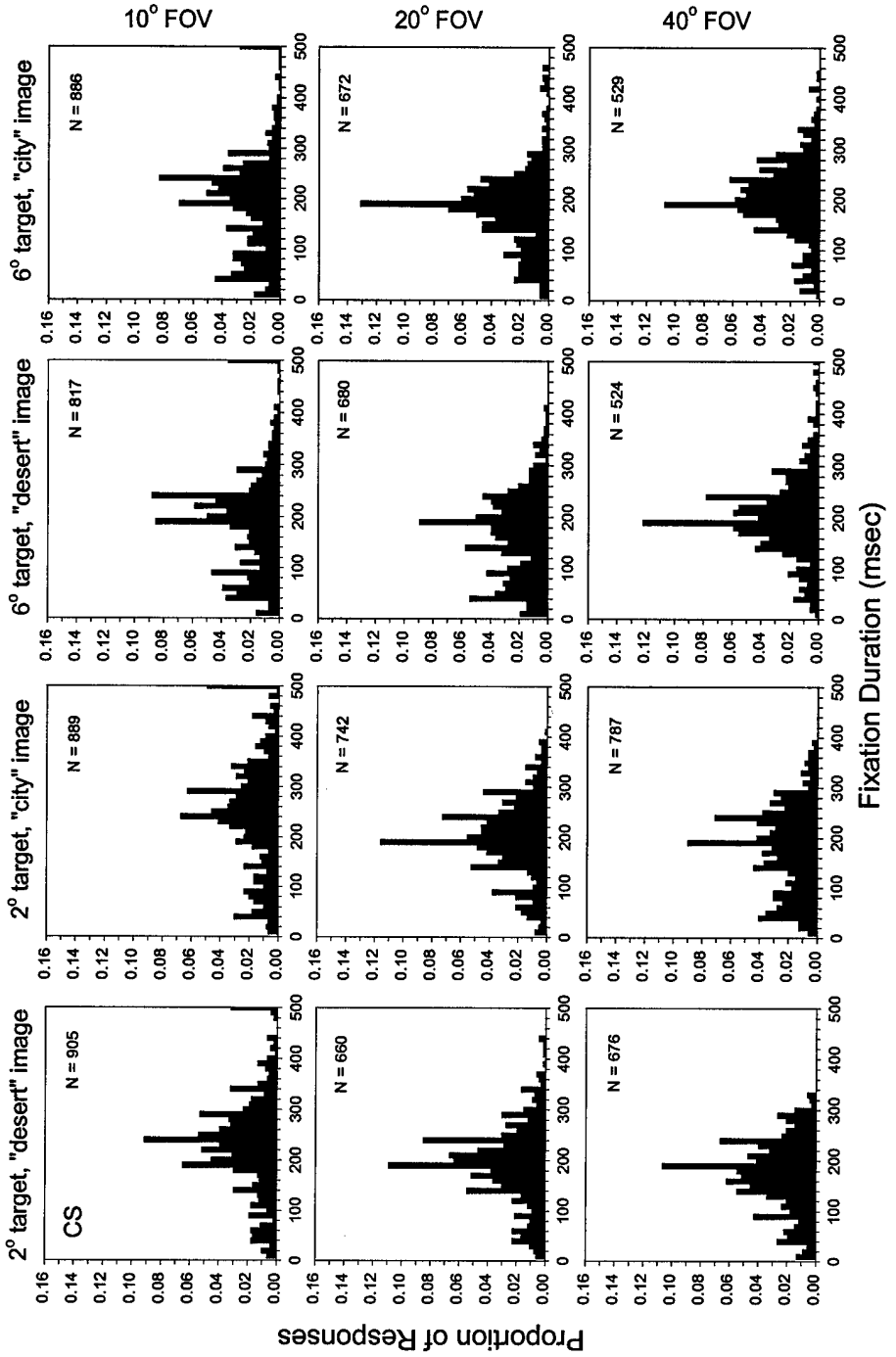

GA, 2° target, BD1
N=709
GA, 2° target, BD2
N=731
GA, 6° target, BD1
N=528
GA, 6° target, BD2
N=676
Proportion of Responses
Gaze Direction (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.20
90.0
180.0
270.0
0.0
90.0

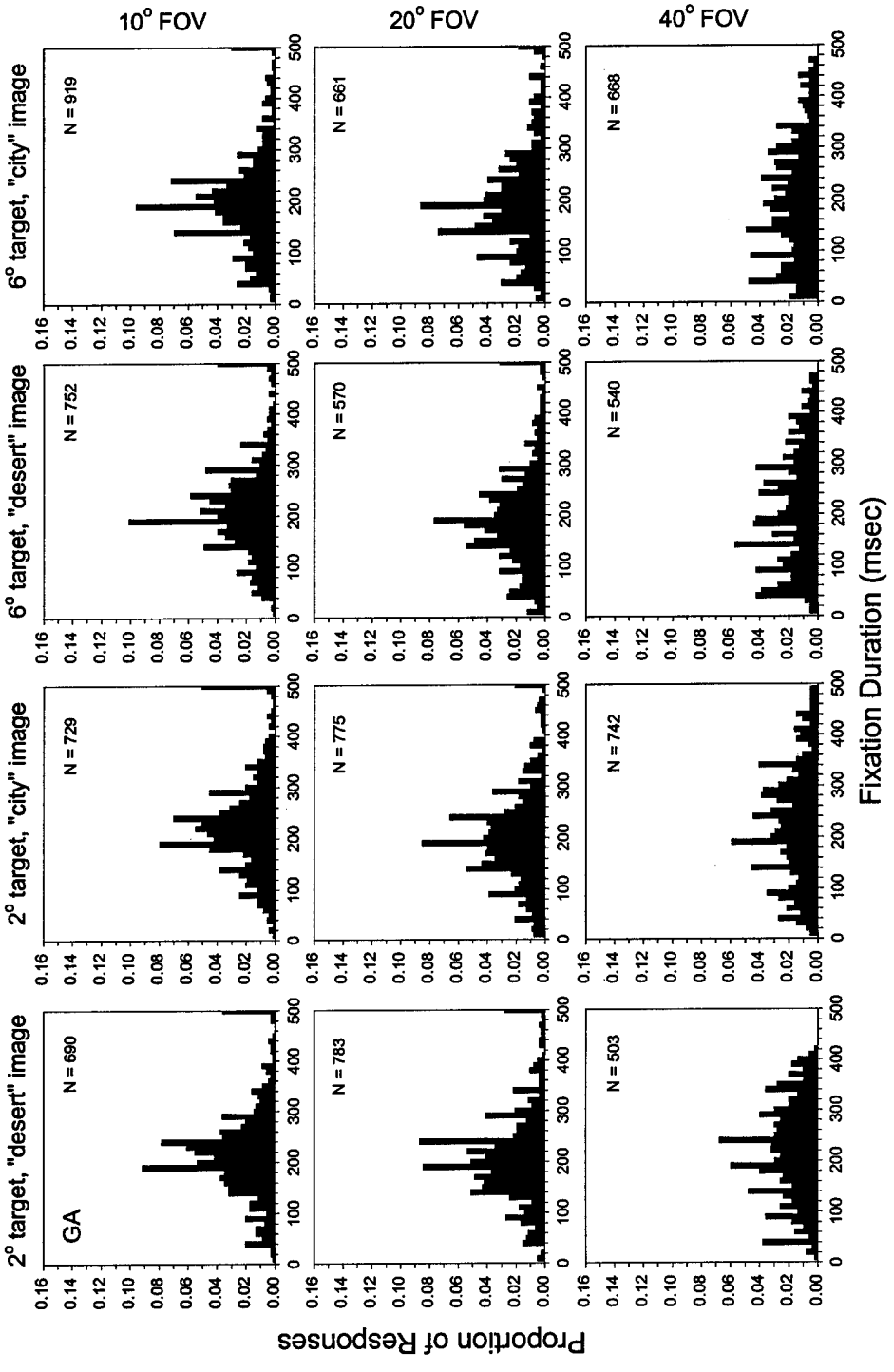

KL, 2° target, BD1
N=605
KL, 2° target, BD2
N=665
KL, 6° target, BD1
N=458
KL, 6° target, BD2
N=495
Proportion of Responses
Gaze Direction (deg)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.20
90.0
180.0
270.0
0.0
90.0

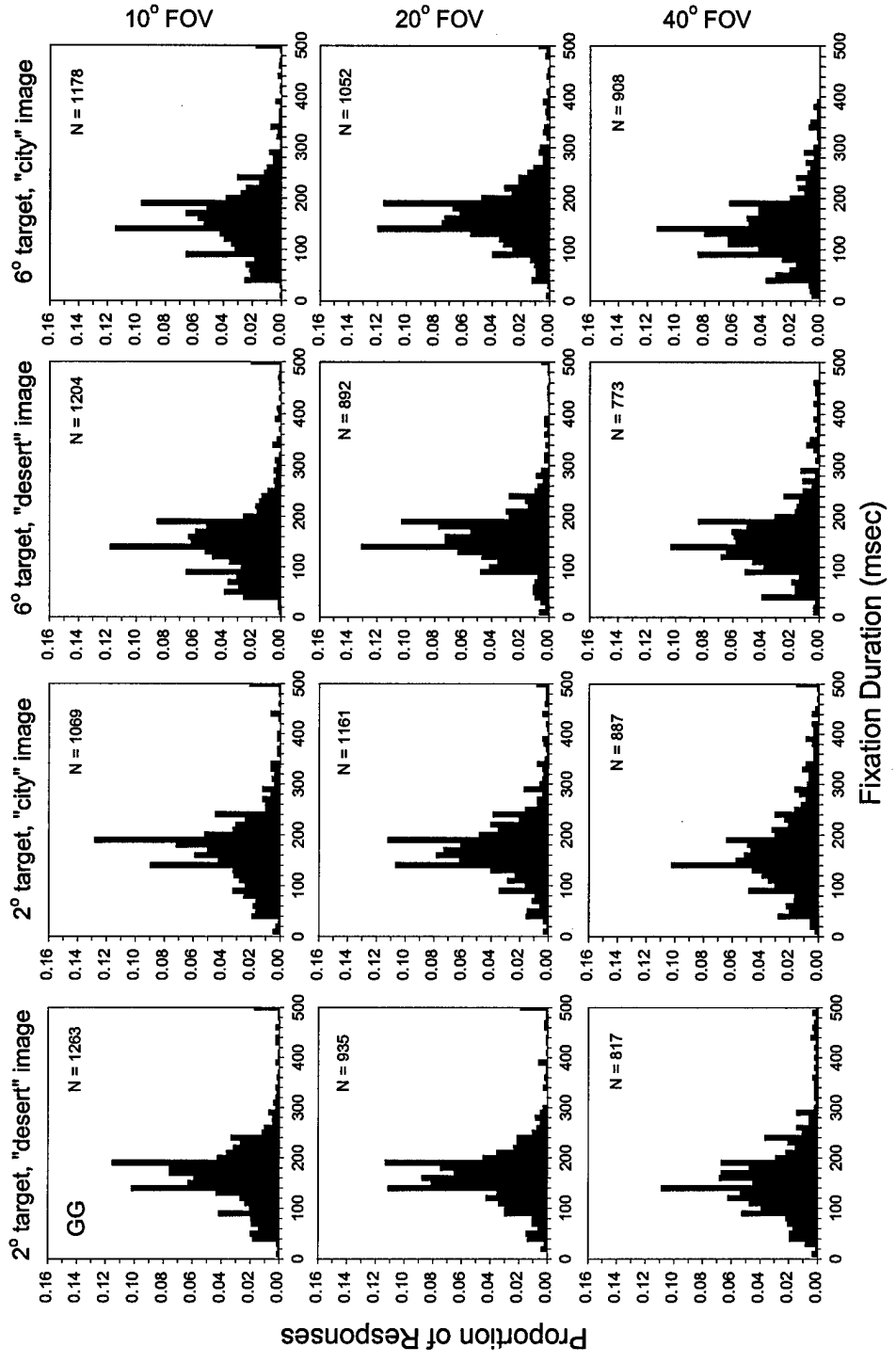
This page intentionally left blank.

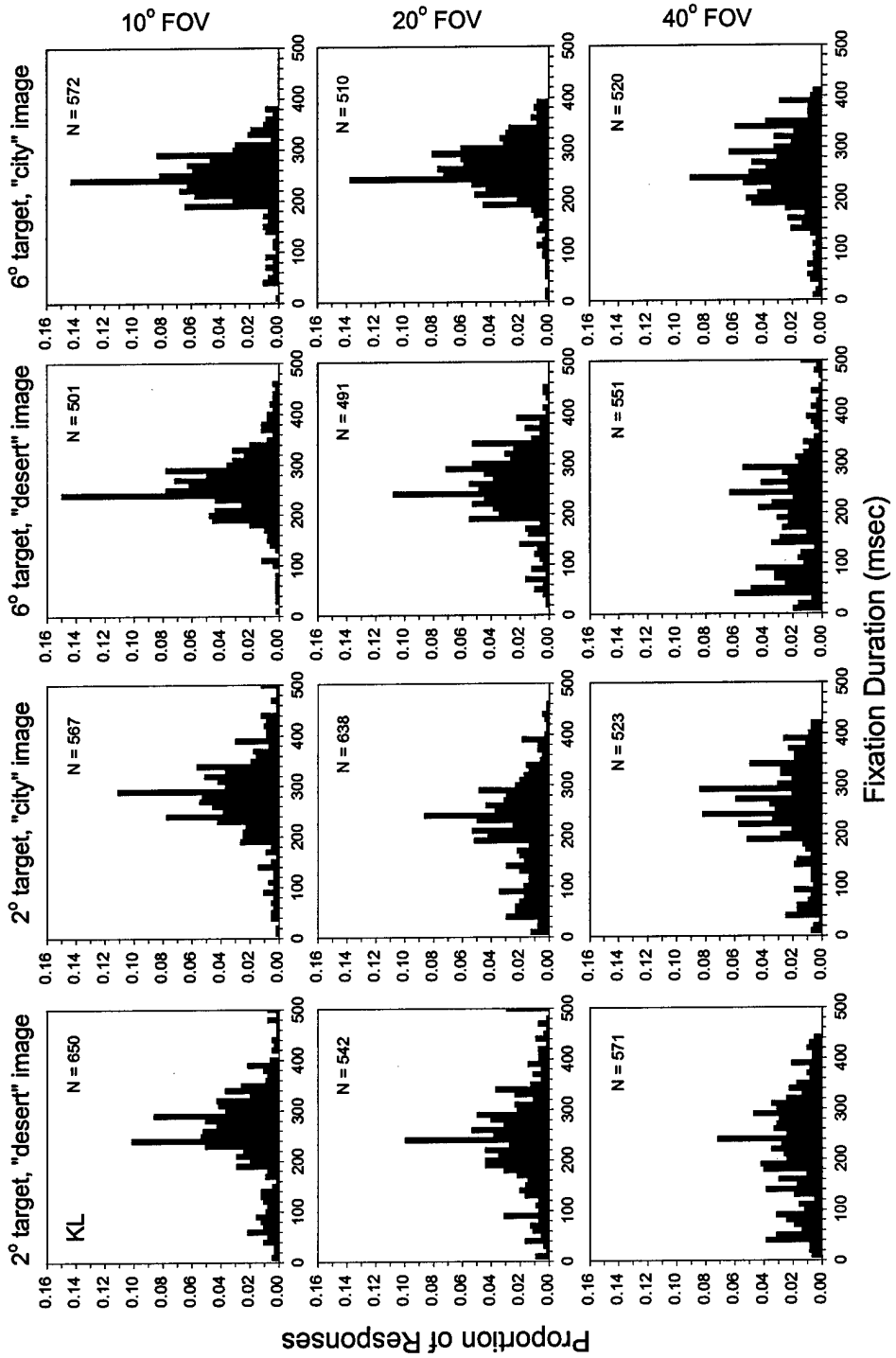
6.6 Appendix F

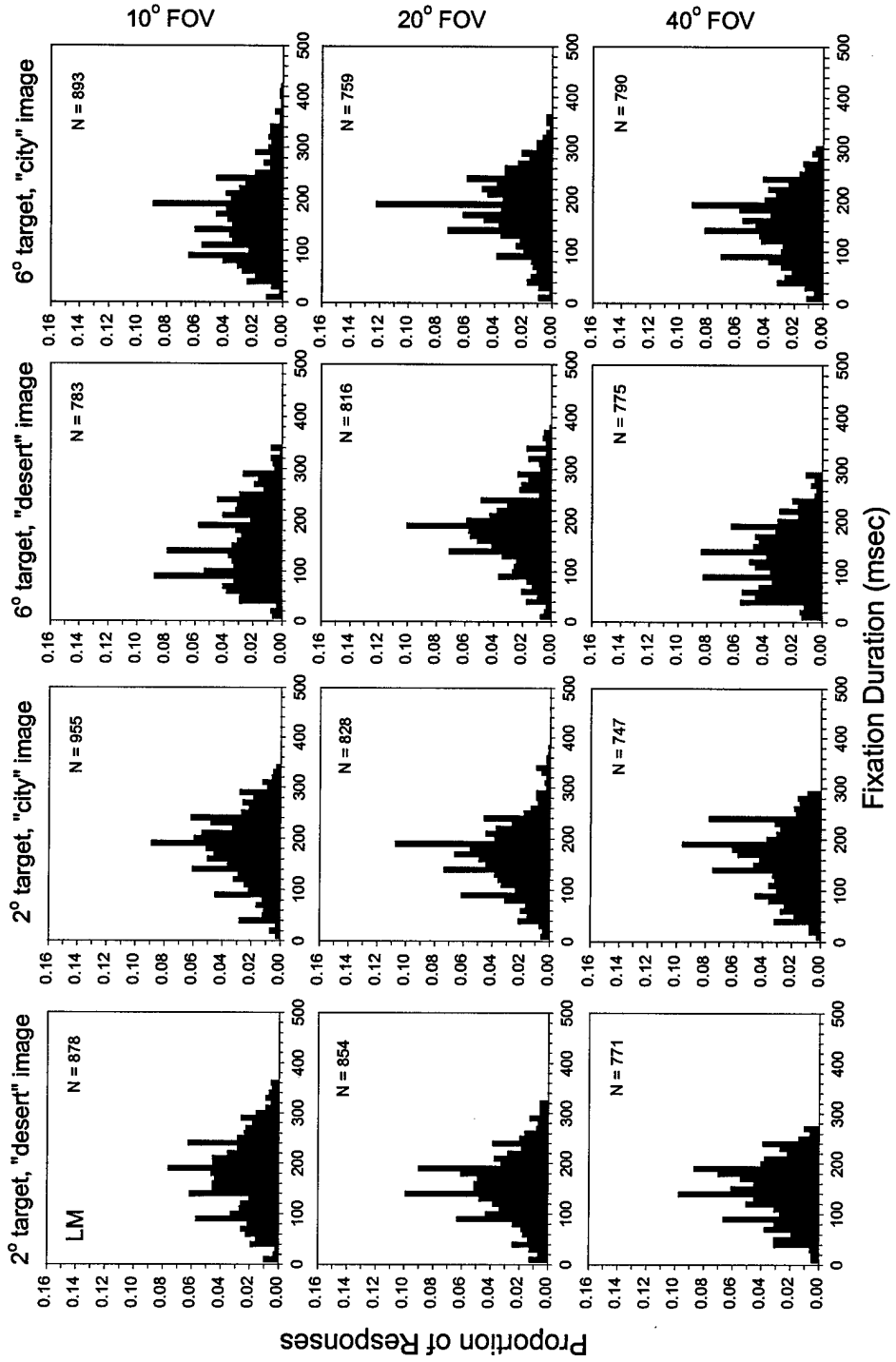
Fixation duration histograms for all observers tested in the IFOV condition.

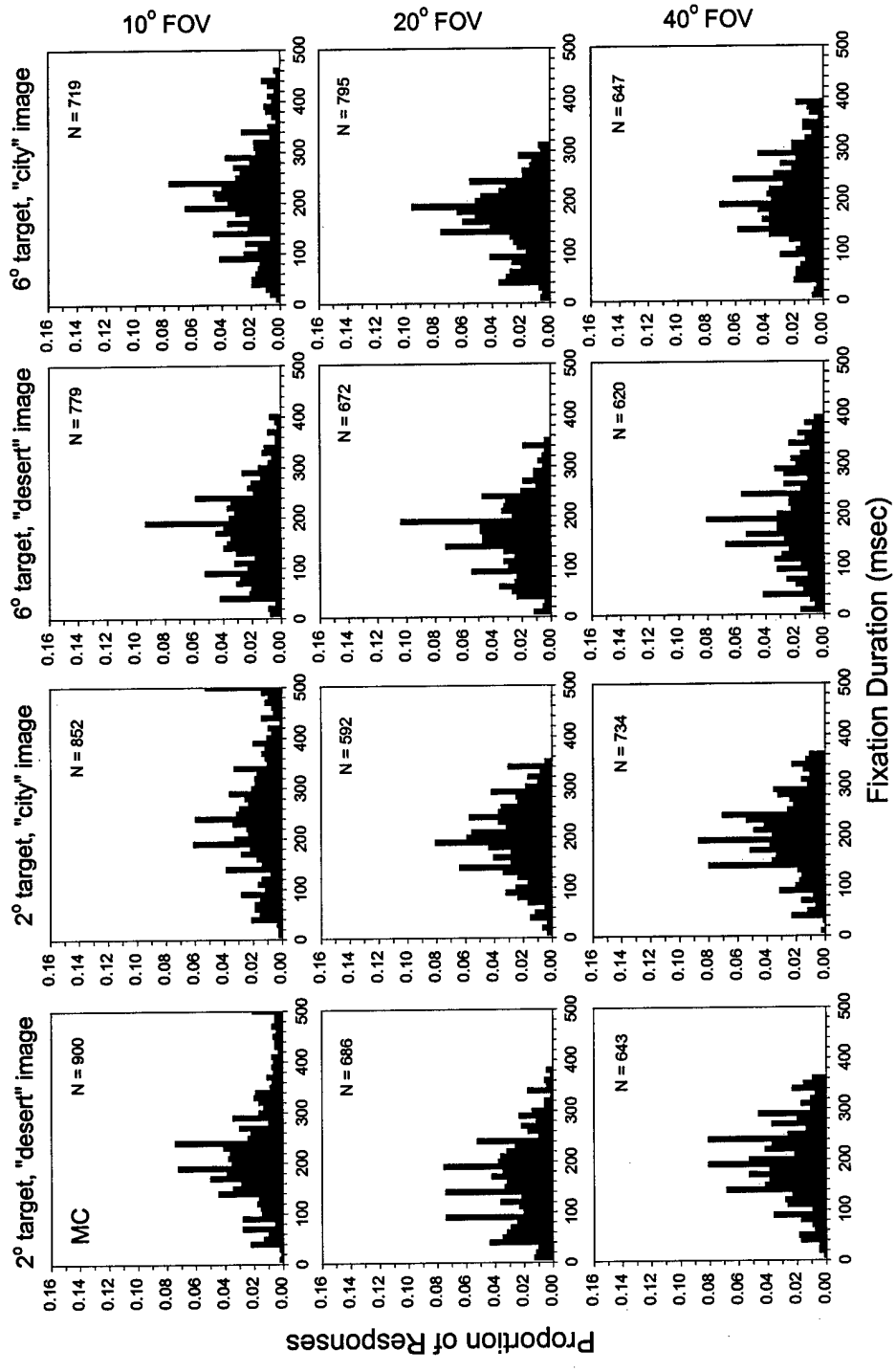

10° FOV
20° FOV
40° FOV
2° target, "desert" image
2° target, "city" image
6° target, "desert" image
6° target, "city" image
CS
N = 905
N = 889
N = 817
N = 886
N = 660
N = 742
N = 680
N = 672
N = 676
N = 787
N = 524
N = 529
Proportion of Responses
Fixation Duration (msec)

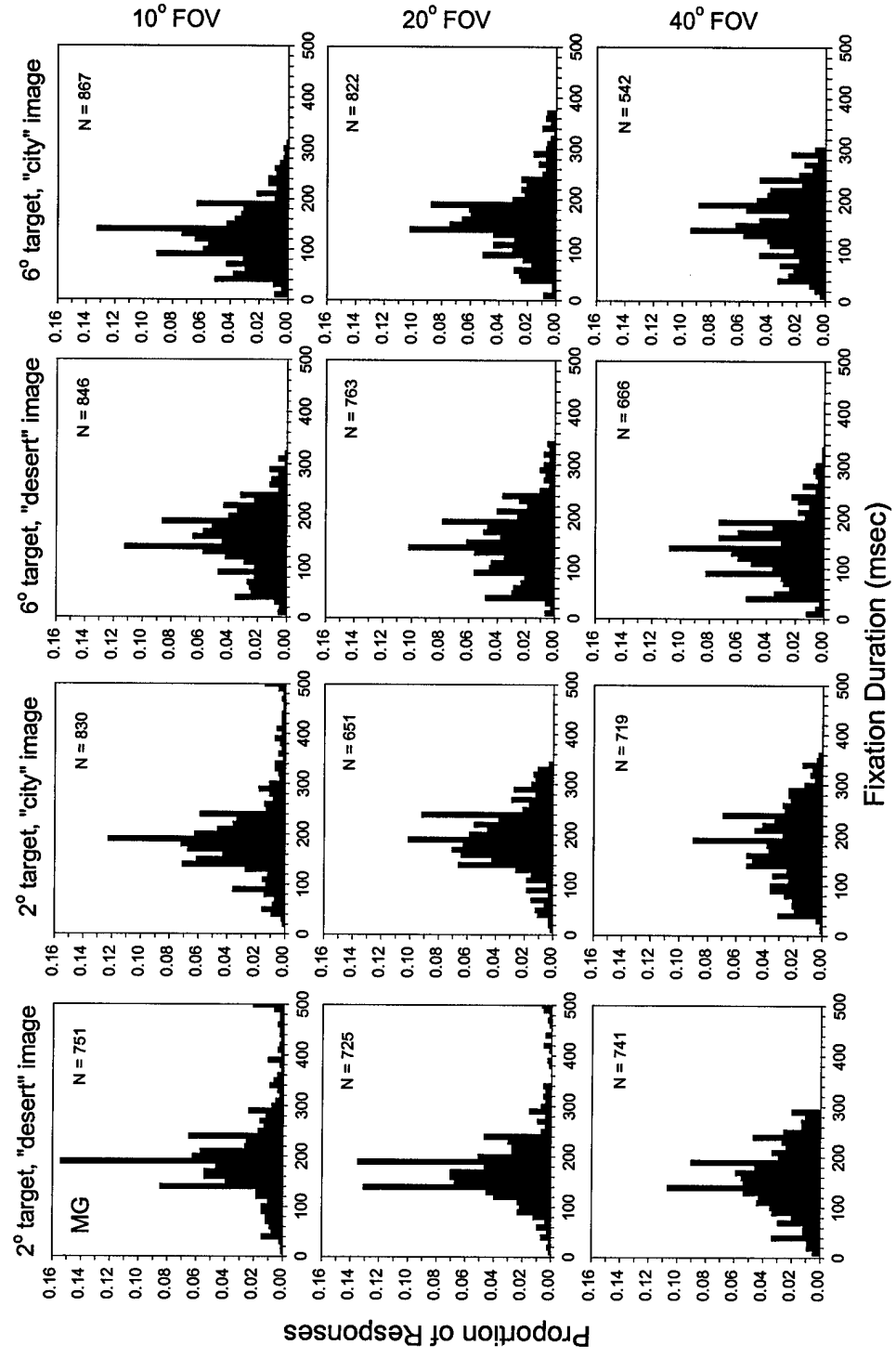

10° FOV
20° FOV
40° FOV
2° target, "desert" image
2° target, "city" image
6° target, "desert" image
6° target, "city" image
GA
N = 690
N = 729
N = 752
N = 919
N = 783
N = 775
N = 570
N = 661
N = 503
N = 742
N = 540
N = 668
Proportion of Responses
Fixation Duration (msec)


10° FOV
20° FOV
40° FOV
2° target, "desert" image
2° target, "city" image
6° target, "desert" image
6° target, "city" image
GG
N = 1263
N = 935
N = 817
N = 1069
N = 1161
N = 887
N = 1204
N = 892
N = 773
N = 1178
N = 1052
N = 908
Proportion of Responses
Fixation Duration (msec)


10° FOV
20° FOV
40° FOV
2° target, "desert" image
2° target, "city" image
6° target, "desert" image
6° target, "city" image
KL
N = 650
N = 542
N = 571
N = 567
N = 638
N = 523
N = 501
N = 491
N = 551
N = 572
N = 510
N = 520
Proportion of Responses
Fixation Duration (msec)


10° FOV
20° FOV
40° FOV
2° target, "desert" image
2° target, "city" image
6° target, "desert" image
6° target, "city" image
LM
N = 878
N = 955
N = 783
N = 893
N = 854
N = 828
N = 816
N = 759
N = 771
N = 747
N = 775
N = 790
Proportion of Responses
Fixation Duration (msec)


10° FOV
20° FOV
40° FOV
2° target, "desert" image
2° target, "city" image
6° target, "desert" image
6° target, "city" image
MC
N = 900
N = 852
N = 779
N = 719
N = 686
N = 592
N = 672
N = 795
N = 643
N = 734
N = 620
N = 647
Proportion of Responses
Fixation Duration (msec)


10° FOV
20° FOV
40° FOV
2° target, "desert" image
2° target, "city" image
6° target, "desert" image
6° target, "city" image
MG
N = 751
N = 830
N = 846
N = 867
N = 725
N = 651
N = 763
N = 822
N = 741
N = 719
N = 666
N = 542
Proportion of Responses
Fixation Duration (msec)